Predicting referential states using enriched texts

Erwin R. Komen

Radboud University Nijmegen and SIL-International
E-mail: E.Komen@Let.ru.nl

Abstract

Information structure research that makes use of diachronic corpora would be greatly facilitated by having texts that are not only syntactically parsed, but for which the referential state (an indicator of newness) of each noun phrase is available as well. Manual or semi-automatic annotation of the referential state is a tedious and time-consuming job, but the availability of 18 texts that have been annotated by our research group, allows training a statistical predictor and evaluating its performance. The statistical predictor discussed in this paper makes use of TiMBL, outperforms a hard-coded deterministic predictor, and reaches an overall precision of 83% to 87%. While this is not good enough for fully automatic annotation of texts, the predictors are usable in the kind of diachronic information structure research that requires only a rough estimate of the referential state of noun phrases.

1 Introduction

Diachronic corpora provide an ideal platform for research into the relation between syntax and information structure, since changes in the syntax of a language may lead to changes in the information structure system, as has been shown for English [10, 16]. Yet, while there is a growing number of historical texts that have been parsed syntactically, which makes them suitable for finding and tracking syntactic changes, texts that contain adequate information structure annotation are scarse. Komen [9] argues that the information structure of a sentence can be calculated from the combination of referential and syntactic information, so that syntactically annotated texts only need to be enriched with referential annotation. It is because of its crucial role in diachronic research that referential enrichment is currently being undertaken by projects like PROIEL [6], ISWOC [1] and the Nijmegen group [8], to which the author belongs.

Referential annotation consists of two parts: (a) a label for the referential status of each relevant constituent, and (b) a link to the antecedent of the constituent, if the constituent has one.[1] The referential status annotation task, if done manually, is a tedious one, since high accuracy is pursued, so that the annotated texts can serve information structure research in the most reliable way.[2] Fully automatic annotation of the five different possible referential states (see 2.1) has not yet been reported, but much work has been done on automatic coreference resolution (part "b" of the referential annotation

[1] The three projects mentioned differ in the particular labels that are used for the referential states (see 2.1).

[2] The Nijmegen group reports a Cohen's kappa above 0.8 for the Pentaset annotation. The Proiel group reports a kappa of 0.89 in distinguishing "New", "Old" and "Accessible", which was reached after several trials and discussions of inconsistently tagged situations [6].

discussed above), even though this generally aims at only finding noun phrases with a "Given" status as well as a link to their antecedent (one exception is [15]). The results of coreference resolution are reported to reach a precision of 70%-81% for the fully automatic task [12, 13]. Information structure research is more interested in part "a": getting the labels for the referential states of NPs. The "Cesax" program handles both referential status and antecedent link, yields high enough precision [8], but its semi-automatic nature requires substantial user-input.

The Nijmegen group has annotated a small body of some 18 texts (appr. 100 kWords) with referential status and antecedent location, and the availability of this material opens the doors to new directions in tackling the annotation task: the annotated texts can be used as training and test data for statistical approaches to the tasks of determining the referential status of constituents and determining their antecedents.[3]

This paper seeks to establish whether a statistical approach to referential status prediction is feasible and how it compares to a deterministic approach. Referential status prediction has applications in those information structure research topics that do not need detailed access to antecedents, but only require coarse-grained referential status distinctions.[4] A full-fledged referential status prediction will also have an application in computational linguistics as a logical preprocessing step before coreference resolution takes place, witness the research conducted by Uryupina [15] and Gegg-Harrison et al. [5].

2 Developing predictors

The way a predictor works depends on the required output (2.1) and the information that is fed into it (2.2). Both the deterministic predictor (2.3) and the statistical predictor (2.4) use the same syntactically annotated texts as input, but their capabilities differ due to their nature: the deterministic one requires detailed prior information about the interaction between syntax and referential states, whereas the statistical one does not.

2.1 The output of the predictor: referential states

The experiments described in this paper use predictors that specify referential state in a number of ways. The coarsest distinction that is made is the one between "Link" constituents and "NoLink" constituents: those marked "Link" have an antecedent in or outside the text, while those that are "NoLink" do not have an antecedent. The finest distinctions that are made

[3] While the principle of using manually annotated material as training for statistical prediction has been used in coreference resolution in general, it has not, as far as I am aware, been used in the finer-grained task of referential annotation.
[4] Research into Old English V2 behaviour and the transition from OV to VO for instance, has, until now, only made use of such coarse information [10, 11, 16].

boil down to five different primitives, that can be referred to as the "Pentaset" [8]. The different referential states can be discerned in the following way:

(1) *Referential states included in "Link" and "NoLink"*

Link

a. Identity (Proiel: OLD) The constituent has an antecedent in the text, and the referents of both are identical.
b. Inferred (Proiel: ACC-inf) The constituent has an antecedent in the text, but the referents of the current constituent and its antecedent are not the same (they can be in a part-whole relation, for instance). The mention of the first noun phrase must already have implied the existence of the second noun phrase, which infers from it.
c. Assumed (Proiel: ACC-sit + ACC-gen) The constituent has an antecedent, but it is outside the text. The referents of the current constituent and this antecedent must be equal.

NoLink

a. New (Proiel: NEW) The constituent does not have an antecedent inside or outside the text, and it can be referred to later on.
b. Inert (Proiel: not labelled) The constituent does not have an antecedent inside or outside the text, and it cannot be referred to in the following context.

Section 3 describes experiments that differ in terms of the output distinctions in referential states that are made. The different schemes are in (2).

(2) *Output schemes for a predictor*

a. "Link-NoLink" – Predict whether the constituent has an antecedent (Link) or not (NoLink).
b. "Link-New-Inert" – Make a three-way distinction. Constituents are first checked on whether they have an antecedent (Link) or not (NoLink). The last category "NoLink" is then further divided in constituents that can function as antecedents (New) and those that cannot (Inert).
c. "Pentaset" – Make the five-way distinction as in (1), so that the output is one of the states of the "Pentaset": Identity, Inferred, Assumed, New or Inert.

The output scheme in (2.a) is chosen, since it seems to be the easiest distinction that can be made and that is still useful for information structure research. The three-way scheme in (2.b) and the five-wey scheme in (2.c) are included to find out in what way increasing the number of referential states to be distinguished influences the performance of the predictors.

2.2 The input to the predictor

The knowledge of which the predictor described in this paper can make use consists of the syntactic, morphological and functional information available in a number of syntactically parsed texts that have been taken of four historical corpora containing excerpts from English literature from roughly

1000 A.D until 1914 (see [14] for the oldest corpus). References to these sources and to the 18 text subset of them that have been enriched with referential status annotation are listed in Komen [9]. The texts are mostly narratives, history and sermons; the average number of words per text is 5000. The referential state annotation that has been added to the noun phrases in these texts uses the five "Pentaset" states discussed in section 2.1. The annotation used in these English corpora differs in detail from the wider-known Treebank II annotation, but the principle is comparable [2, 14].

A referential state predictor needs to be able to determine the status of all noun phrases in a text, except for those that are lexically empty (their antecedents are predictable and including them would skew the data).[5] The information a referential state predictor is able to use, then, consists of all the morphological, syntactic and functional information available in the parsed English corpora.

2.3 The deterministic predictor

The Deterministic predictor is hard-coded as a built-in Xquery function `ru:RefState` within the program "CorpusStudio" [7]. It is within this program that it has easy access to all kinds of information that can be gleaned from the syntactically parsed texts. The output of the deterministic predictor is the two-way Link-NoLink division. A description of the algorithm behind `ru:RefState` follows here in (3).

(3) REFSTATE(ndThis)

```
1  hd ← HEAD(ndThis)
2  npt ← hd.NPtype
3  if npt = 'Proper' then
4      return (if OCCURSBEFORE(hd) then 'Link' else 'NoLink')
5  end if
6  pm ← POSTMODIFIER(ndThis, hd)
7  if EXISTS(pm) then
8      if hd.Label in Adj, Adv, N, NS, Num, PP, Q return 'NoLink'
9      else if hd.Label in Pro, D return 'Link'
10     else return 'Link'
11     end if
12 end if
13 if npt in Dem, DemNP, Pro, PossPro, PossDet, … return 'Link'
14 else return 'NoLink'
15 end if
```

The algorithm first of all in (3.1) attempts to find the head of the NP. If the head is a proper noun (3.3), a previous occurrence determines whether the referential state prediction is "Link" or "NoLink" respectively. Step (3.6) of the algorithm checks the presence of a post-modifier, and if it finds one, it

[5] Lexically empty NPs in the parsed English corpora include subjects that are elided under coordination, traces for *wh*-movement, empty expletives and some more categories [14].

determines the referential status just by looking at the syntactic tag of the head in (3.8-11). If there was no post-modifier, then steps (3.13-15) of the `ru:RefState` procedure evaluate the "NPtype", which comes in the form of a feature that has been automatically added to every noun phrase previously, solely on the basis of the available syntactic information.[6] Certain types of noun phrases translate directly into a matching referential states: noun phrase types of DEM (independent demonstrative pronouns), DEMNP (noun phrases headed by a demonstrative), PRO (pronouns), POSSPRO (possessive pronouns) and POSSDET (noun phrases that start with a determiner in the form of a possessive noun or proper noun) all result in a state of "Link", and noun phrases of type QUANTNP (quantifier noun phrases), INDEFNP (indefinite NPs), BARE (bare nouns) and EXPL (explitives) are all marked as "NoLink".

2.4 The statistical approach

The statistical predictor that has been used in this research makes use of memory-based learning (see [3] for an introduction in memory-based language processing). The reason to opt for a memory-based approach instead of for a more generalizing approach such as a maximum entropy one or a naive Bayesian, is that it seems quite likely that there are some idiosyncratic combinations of features determining a particular referential state, and we also expect there to be lexical dependencies. A statistical approach that defines classes based on generalizing over samples will, necessarily, miss out on idiosyncratic outcomes, whereas a memory-based approach should not.

The memory-based approach, being statistical in nature, needs to start with a training phase: it needs to have a collection of feature-value combinations with their corresponding classification. The input for the *training* consists of a list that gives the features of each noun phrase and the referential status that has been assigned to it. The referential status of a newly encountered noun phrase is, after training has taken place, determined by comparing the features of this noun phrase with the features of *all* the noun phrases in the training set. The referential category of the nearest neighbour in the feature space is assigned to this new noun phrase. Table 1 lists the features that have been used in the three experiments described in this paper.

The information available in the five features used in experiment 1 is comparable to what the function `ru:RefState` in the deterministic approach described in 2.2 uses: the information contained in the label of the NP (feature "NP_Label"), the information available from the head (feature "Head_Label"), the presence of an anchor (feature "Ch_Anchor"), the post-modifiers information (feature "Ch_PostMod") and the value of the NPtype

[6] The "NPtype" can have the following values: DEM, DEMNP, PRO, POSSPRO, PRONP, POSSDET, QUANTNP, INDEFNP, BARE, BAREWITHPP, EXPL, ANCHOREDNP, PROPER, DEFNP, FULLNP or UNKNOWN. The exact definition of these categories is less important for the purpose of this paper.

feature (feature “NP_Type”). The difference between the two methods is that the memory-based approach does not depend upon hard-coding, which means that it is not prone to oversight on the part of the programmer, and that it is easily extendable to other languages, without requiring language-specific knowledge.

Name	**Description**	**Exp 1**	**Exp 2**	**Exp 3**
Period	Time-period of text		+	+
NP_Label	Phrase label of NP	+	+	+
NP_Type	NP feature	+	+	+
NP_GrRole	NP feature		+	+
NP_PGN	NP feature		+	+
NP_words	Number of words in NP		+	+
Ch_FreeRel	NP is a free relative		+	+
Ch_Rel	NP has an RC child		+	+
Ch_Neg	NP has a negator		+	+
Ch_PreMod	Phrase label of pre-modifier		+	+
Ch_PostMod	Phrase label of post-modifier	+	+	+
Ch_Anchor	NP has a possessive pronoun	+		
Ch1_Label	Phrase label of NP-child #1		+	+
Ch1_WrdType	Word type of NP-child #1		+	+
Ch1_WrdText	Text of NP-child #1		+	+
Ch2_Label	Phrase label of NP-child #2		+	+
Ch3_Label	Phrase label of NP-child #3		+	+
Ch4_Label	Phrase label of NP-child #4		+	+
Head_Text	NP-head text		+	+
Head_Before	NP-head occurred earlier		+	+
Head_Label	NP-head phrase or POS label	+		
SisterBE	NP has *be*-verb sister		+	+
SisterSBJ	NP has subject as sister		+	+
SisterV	NP has verbal sister		+	+
SisterCP	NP has any CP as sister		+	+
Sbj_NPtype	NPtype feature of subject		+	+
Sbj_Text	Text of the subject		+	+
Cls_Mood	Mood of the clause		+	+
Cls_Speech	Clause is direct speech		+	+

Table 1 Features used in the memory-based approach[7]

Experiments 2 and 3 make use of a larger feature set, as shown in Table 1, but they do away with the Ch_Anchor feature (which is replaced by the more general Ch1_Label feature) and the Head_Label feature.

---

[7] There is a lot of ‘implicit’ information in the features. NPs that are non-linking since they are part of a presentational constructions of type “there is/are NP”, for instance, can be recognized by the combination of “SisterSBJ” having value “1” and the feature “Sbj_NPtype” having the value “Expletive”.

3 Results

Both the deterministic as well as the statistical predictor are evaluated by making use of the “CorpusStudio” program, which provides an environment for running Xquery on the annotated English texts [7].

3.1 Testing the performance of the predictors

The deterministic referential state predictor does not require learning, since it is “hard-wired” as the Xquery function `ru:RefState` within the program “CorpusStudio”. The Xquery code that uses this function and returns a list outlining how many instances of each of the predicted states have been found for each of the actual states follows the algorithm sketched in (4).

(4) TESTPREDICTOR(*list_of_texts*, *pred_type*, *scheme*)

```
1  for each np in list_of_texts
2      if not(empty(np)) then
3          ref ← np.RefState
4          actual ← SCHEMESTATE(ref, scheme)
5          if pred_type = ‘Deterministic’ then
6              pred ← RU:REFSTATE(np)
7              ADDTOOUTPUT(actual, pred)
8          else
9              feat_vec ← GETFEATUREVECTOR(np)
10             TIMBLPREP(feat_vec, actual, 70)
11     end if
12 next np
```

The algorithm in (4), when called with *pred_type* set to ‘Deterministic’, serves to test the “Link-NoLink” output scheme described in (2) in section 2.1. The deterministic predictor (see section 2.2 and the algorithm in 3) in the form of the function `ru:RefState` does not distinguish all five referential states from the Pentaset, which means that we cannot test its performance for the second and third predictor output scheme (2.b) and (2.c).

Since the memory-based predictor is a statistical one, testing is done by dividing the available data into a training set and a test set. The procedure to test the memory-based predictor retrieves the actual referential state of each noun phrase, it determines the values of the features that are going to be used in the prediction, and it divides the noun phrases of the 18 texts used for this experiment over a training and a test set.

The variable *scheme* that is passed to the procedure in (4) determines which of the three output schemes is required: (a) the two-way “Link-NoLink” division, (b) the three-way “Link-New-Inert” division, or (c) the five-way Pentaset division “Identity-Inferred-Assumed-New-Inert”. Step (4.4) makes sure the output state matches the scheme that is being used. Step (4.9) calls a user-built Xquery function that determines the values of the features used for the predictor, and step (4.10) makes sure that a training set

(70%) and a test set (30%) with feature vectors and outcomes is prepared for further processing by TiMBL, the memory-based engine that is used here [4].[8] It is TiMBL that performs the actual prediction.

3.2 Performance of the two predictors

So far, I have discussed two predictors, a deterministic one (2.2) and a statistical one (2.4), and I have shown that there are slightly different ways to test the performance of these predictors (3.1), although they both can be tested by using the program CorpusStudio. This section discusses the outcome of the tests that have been done to determine the performance of the predictors, and it follows the predictor outcome schemes described in (2).

3.2.1 Performance for the "Link-NoLink" output

The first comparison of the two different predictor types takes the "Link-NoLink" output scheme defined in (2.a). The deterministic predictor is tested on all of the available data, while the statistical one is trained on 70% of the data and then tested on the remaining 30%. Table 2 contains the confusion matrices that show which kinds of mistakes are being made by the predictors, as well as the overall performance in terms of precision and F-score (abbreviated as "P, F").

	Deterministic predictor				**Memory-based (timbl) predictor**			
Actual	**Link**		**NoLink**		**Link**		**NoLink**	
Link	**9977**	41,60%	2867	11,90%	**3320**	47,00%	472	6,70%
NoLink	1089	4,50%	**10049**	41,90%	445	6,30%	**2825**	40,00%
P, F	83,5%, 91,0%				87,0%, 93,1%			

Table 2 Predictor performance for the "Link-NoLink" output scheme

The statistical memory-based predictor outperforms the deterministic one for this test: the precision is higher and the F-score is slightly higher too. The performance of the predictors for individual referential outcome categories can best be observed by looking at their individual precision, recall and F-score values.

Method	**Feature**	**TP**	**FP**	**FN**	**TN**	**Precision**	**Recall**	**F-Score**
Deterministic	Link	9977	1089	2869	10057	90,2%	77,7%	83,4%
Deterministic	NoLink	10049	2868	1096	9979	77,8%	90,2%	83,5%
Statistical	Link	3320	445	472	2825	88,2%	87,6%	87,9%
Statistical	NoLink	2825	472	445	3320	85,7%	86,4%	86,0%

Table 3 Performance per referential state for the "Link-NoLink" scheme

The statistical memory-based predictor shows a more balanced behaviour when it comes to the performance on individual referential states. The

[8] The memory-based predictor uses the default settings of TiMBL: "IB1" algorithm, "Overlap" metric, "GainRatio" weighing. Future work will include experimenting with different settings, in order to see how much the predictor can be improved.

deterministic one is particularly bad at predicting the state “NoLink”: there are over twice as much false-positives (marked “FP”) than when predicting the “Link” state. The deterministic predictor gains a higher precision when it comes to predicting the state “Link”, but it does so at the cost of an increased number of false-negatives (marked “FN”), which leads to a smaller recall.

Apparently the deterministic predictor is too conservative in recognizing the state “NoLink”, while it is too optimistic when assigning the state “Link”. A detailed analysis of the data would be needed to find out exactly why this is so, and what could be done to remedy this.

3.2.2 Performance for the “Link-New-Inert” output

While the deterministic predictor is limited to discerning whether an NP has a “Link” or a “NoLink” state, the statistical predictor can be used to get a more detailed output. The second experiment for the referential state prediction is done only with the statistical predictor, it uses the 27 features shown in Table 1, and its outcome distinguishes three states: (a) “Link” (which combines the Pentaset states of “Identity”, “Inferred” and “Assumed”), (b) “New” and (c) “Inert”. Table 4 shows the confusion matrix that results for this output scheme.

Actual	**New**		**Link**		**Inert**	
New	**2387**	33,1%	404	5,6%	127	1,8%
Link	380	5,3%	**4065**	56,4%	43	0,6%
Inert	114	1,6%	58	0,8%	**488**	6,8%
P, F	86,0%, 92,5%					

Table 4 Predictor performance for the “Link-New-Inert” output scheme

Comparing the confusion matrix in Table 4 with the one in Table 2, we can see that the overall performance of the statistical predictor does not change radically when we turn from a rough two-way distinction (87% precision) to a finer three-way one (86%). The performance of the three different states is shown in Table 5.

Feature	**TP**	**FP**	**FN**	**TN**	**Precision**	**Recall**	**F-Score**
New	2387	494	531	5148	82,9%	81,8%	82,3%
Link	4065	462	423	3578	89,8%	90,6%	90,2%
Inert	488	170	172	7406	74,2%	73,9%	74,1%

Table 5 Performance per referential state for the “Link-New-Inert” scheme

The precision of the state “Link” in Table 5 (89,8%) is actually a little better than the one for “Link” in the two-way experiment in Table 3 (88,2%), which is what we would expect, given the higher number of features (27 instead of 5) taken into account. Making the distinction between the referential states “New” and “Inert” proves to be possible with less precision, and future work will need to find out whether crucial features are missing from the set that make the prediction of these states more effective.

3.2.3 Performance for the "Pentaset" output

The third experiment aims at predicting the full range of Pentaet states (2.c), which is currently only possible with the statistical predictor. Table 6 shows that the overall precision does decrease for this task (it changes from 86% in the three-way distinction to 81% in the five-way Pentaset distinction), and Table 7 shows the performance of the predictor for each of the individual referential states.

Actual	**Assumed**		**New**		**Identity**		**Inferred**		**Inert**	
Assumed	**63**	0,9%	63	0,9%	82	1,1%	20	0,3%	0	0,0%
New	42	0,6%	**2360**	32,7%	228	3,2%	121	1,7%	116	1,6%
Identity	57	0,8%	210	2,9%	**3474**	48,2%	87	1,2%	32	0,4%
Inferred	13	0,2%	105	1,5%	158	2,2%	**87**	1,2%	11	0,2%
Inert	3	0,0%	97	1,3%	44	0,6%	12	0,2%	**433**	6,0%
P, F					81,0%, 89,5%					

Table 6 Predictor performance for the "Pentaset" output scheme

Feature	**TP**	**FP**	**FN**	**TN**	**Precision**	**Recall**	**F-Score**
Assumed	63	115	165	7575	35,4%	27,6%	31,0%
New	2360	475	507	4576	83,2%	82,3%	82,8%
Identity	3474	512	386	3546	87,2%	90,0%	88,6%
Inferred	87	240	287	6871	26,6%	23,3%	24,8%
Inert	433	159	156	7170	73,1%	73,5%	73,3%

Table 7 Performance per referential state for the "Pentaset" output scheme

The precision for determining the referential states "Assumed" and "Inferred" are both quite low: 35,4% and 26,6% respectively. Only 87 out of a total of 374 noun phrases that should be labelled "Inferred" are recognized as such. The reason for these mis-classifications may be quite obvious: the form of a noun phrase alone (e.g. *the table*) is just not enough to be able to say whether it has the referential state "Identity", "Assumed" or "Inferred": in all three situations definite NPs may occur. The feature "Head_Before" makes it clear whether the current NP has in some context been mentioned previously in the text, and in this way helps recognizing clear "Identity" cases. But it is quite obvious that more research needs to be done to find relevant features that help distinguish "Assumed" and "Inferred" NPs.

4 Conclusions and discussion

In this paper I have described and evaluated the feasibility of a statistical approach to referential status prediction and I have compared it with a deterministic approach. The deterministic predictor does not need training, but is very much language (and corpus) specific. The statistical predictor discussed here makes use of memory-based learning, needs training, but outperforms the deterministic one. The evaluation has consisted of three experiments, and the first one was a direct comparison between the two predictors, where both were fed with more or less the same information, and where the outcome was a two-way referential state division: “Link” versus “NoLink”. The deterministic predictor reached an overall precision of 83,5%, while the statistical one reached 87%. The second and third experiments concentrated on establishing the limits of the statistical predictor, which was now fed with 27 features. The outcome of the second experiment was a three-way referential state division, Link-New-Inert, and the precision reached was 86%. The precision with which the three states were predicted did not differ very much. The outcome of the third experiment was a full-fledged five-way referential state division, and the precision reached was 81%. This experiment revealed the current limit in referential state prediction: “Assumed” and “Inferred” were not predicted with an acceptable precision and recall. Future work will aim for a detailed investigation of the circumstances under which these referential states occur, in order to improve the precision of their prediction. Referential state prediction is a logical complementary task to coreference resolution, and its potential as a preprocessing step deserves more attention.

To sum up, referential state prediction is possible, and ready to use for coarse-grained diachronic research, there is room for improvements and it seems likely to serve as sparring partner for coreference resolution.

5 References

[1] Bech, Kristin, and Eide, Kristine Gunn (2011) The annotation of morphology, syntax and information structure in a multilayered diachronic corpus. *Journal for language technology and computational linguistics* 26, (2), 13-24.

[2] Bies, Ann, Ferguson, Mark, Katz, Karen, and MacIntyre, Robert (1995) Bracketing guidelines for Treebank II style Penn Treebank project

[3] Daelemans, Walter, and Bosch, Antal van den (2005) Memory-based language processing (Cambridge University Press, 2005)

[4] Daelemans, Walter, Zavrel, Jakub, van der Sloot, Ko, and Bosch, Antal van den (2009) TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide ILK Technical Report 10-01.

[5] Gegg-Harrison, Whitney, and Byron, Donna K. 2004 Eliminating non-referring noun phrases from coreference resolution. In Proc. Proceedings of DAARC, pp. 21-26

[6] Haug, Dag T. T., Jøhndal, Marius L., Eckhoff, Hanne M., Welo, Eirik, Hertzenberg, Mari J. B., and Müth, Angelika (2009) Computational and Linguistic Issues in Designing a Syntactically Annotated Parallel Corpus of Indo-European Languages. *TAL* 50, (2), 17-45.

[7] Komen, Erwin R. (2009) CorpusStudio. Nijmegen: Radboud University Nijmegen.

[8] Komen, Erwin R. (2012) Coreferenced corpora for information structure research. In Tyrkkö, Jukka, Kilpiö, Matti, Nevalainen, Terttu, and Rissanen, Matti (eds.) *Outposts of Historical Corpus Linguistics: From*

the Helsinki Corpus to a Proliferation of Resources. (Studies in Variation, Contacts and Change in English 10). Helsinki, Finland: Research Unit for Variation, Contacts, and Change in English.

[9] Komen, Erwin R. (2013): Finding focus: a study of the historical development of focus in English. PhD dissertation, Radboud University Nijmegen

[10] Los, Bettelou, López-Couso, María José, and Meurman-Solin, Anneli (2012) On the interplay of syntax and information structure. In Meurman-Solin, Anneli, López-Couso, María José, and Los, Bettelou (eds.) *Information structure and syntactic change in the history of English*, pp. 3-18. New York: Oxford University Press.

[11] Pintzuk, Susan, and Taylor, Ann (2006) The loss of OV order in the history of English. In van Kemenade, Ans, and Los, Bettelou (eds.) *The Blackwell Handbook of the History of English*. Oxford: Blackwell.

[12] Poon, Hoifung, and Domingos, Pedro 2008 Joint unsupervised coreference resolution with Markov logic. In Proc. EMNLP '08 Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 650-659

[13] Raghunathan, Karthik, Lee, Heeyoung, Rangarajan, Sudarshan, Chambers, Nathanael, Surdeanu, Mihai, Jurafsky, Dan, and Manning, Christopher 2010 A multi-pass sieve for coreference resolution. In Proc. EMNLP '10 Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 492-501

[14] Taylor, Ann (2003) The York-Toronto-Helsinki Parsed Corpus of Old English Prose *Syntactic Annotation Reference Manual* University of York.

[15] Uryupina, Olga (2009) Detecting anaphoricity and antecedenthood for coreference resolution *Procesamiento del lenguaje natural. N. 42 (marzo 2009)*, pp. 113-120. Jaén: Sociedad Española para el Procesamiento del Lenguaje Natural.

[16] van Kemenade, Ans, and Milicev, Tanja (2012) Syntax and discourse in Old English and Middle English word order. In Jonas, Dianne, Garrett, Andrew, and Whitman, John (eds.) *Grammatical Change: Origins, Nature, Outcomes*, pp. 239-254. Oxford: Oxford University Press.