NP Modifier Generation Based on Corpus Annotation

Hua Cheng and Massimo Poesio and Renate Henschel and Chris Mellish
Division of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh EH9 8PX, UK
huac,poesio,henschel@cogsci.ed.ac.uk, chrism@dai.ed.ac.uk

Paper ID: NAACL-2001-000

Keywords: non-referring NP modifier, corpus annotation, natural language generation

Contact Author: Hua Cheng

Under consideration for other conferences (specify)? none

Abstract

This paper describes the derivation of preferences for the use of non-referring NP modifiers by annotating a corpus of museum descriptions. It analyses the achieved reliabilities in annotating the semantic and pragmatic features of such modifiers and the regularities in their selection and realisation. These observations are implemented in the ILEX system.

NP Modifier Generation Based on Corpus Annotation

Abstract

This paper describes the derivation of preferences for the use of non-referring NP modifiers by annotating a corpus of museum descriptions. It analyses the achieved reliabilities in annotating the semantic and pragmatic features of such modifiers and the regularities in their selection and realisation. These observations are implemented in the ILEX system.

1 Introduction and Motivation

1.1 Generating Complex NPs

Noun phrase (NP) is an important construction in languages and is often the major means for carrying information in a text. It is generally agreed that the primary function of an NP is to denote concepts, in particular, to refer to concrete discourse entities in a domain. In addition, an NP can serve other communicative goals such as providing new information about the referent and expressing the speaker's emotional attitude towards the referent (Appelt, 1985). In the following example, the italic part refers to an object in a museum, and the bold-faced part provides additional information about it.

This example ***from the time of the Qianlong Emperor 1736-95*** is made of lacquered wood with decoration in gold and red.

Such complex NPs appear frequently in human written texts. A natural language generation (NLG) system must be able to produce complex NPs serving multiple goals in order to write texts as human does. In this paper, we assume that each NP denotes an object which can be a concrete physical object or an abstract concept that denotes a class of objects or something that only exists in human mind. We divide the components of an NP into two parts based on the different functions/communicative goals they serve:

- *a referring part*: intends to refer to an object, but not necessarily to identify, that is, the expression denotes an individual object of a certain class, but it might not be necessary to know the exact object. For example, the italic part of the above example is the referring part of that NP. This part is normally the task of the NP generation module of an NLG system.
- *a non-referring part*: intends to provide additional information about the referent denoted by the referring part, e.g. the bold-faced part of the above example. If an NLG system generates such modifiers, it usually does this in its aggregation module.

This division is more of a functional one than a syntactic one. Except for the head and the determiner, which are always members of the referring part, other syntactic slots can belong to different parts in different circumstances. A referring part mainly serves the referring function, but it may also inform properties of the referent. A non-referring part only serves the informing function and is optional.

In NLG, research on NP generation focuses on deciding syntactic forms and choosing disambiguating modifiers, e.g. (Dale, 1992; Horacek, 1995). Other types of modifiers have not been given sufficient attention. Work on aggregation is satisfied with devising a few rules to allow some degrees of embedding, which is a subprocess of aggregation to generate subordinate NP components. There lacks in-depth discussion of the problem of generating non-referring modifiers in general. It has been argued that generating such modifiers is not a trivial decision because it interferes with the planning of both local and global coherence (in the sense of (Grosz and Sidner, 1986)) (Cheng and Mellish, 2000a). NP subordination, as an alternative to sententialisation, is a major way to achieve conciseness as it reduces the number of sentences in generated texts. However, generating arbitrarily complex NPs is not often desirable, that is, achieving coherence and conciseness are two conflicting considerations and conciseness should only be satisfied under coherence constraints.

This paper focuses on the generation of non-referring NP modifiers in museum descriptions, which provide information for items displayed in a museum. Through analysing descriptions written and revised by human experts, we wish to find out general rules for the usage of such modifiers in the museum domain and hopefully they are general enough to be applied to other domains. Since what we will study are coherent human texts, the NP modifiers in them are produced under both coherence and conciseness considerations.

1.2 Corpus Analysis for NLG

Corpus analysis has often been used for devising rules for modifier generation. For example, Robin (1994) analyses a corpus of reports for basketball matches, which leads to a set of revision rules for generating complex head sentences containing complex NP structures as well as a revision-based generation architecture. Shaw and McKeown (1997) analyse a corpus of patient information to find out regularities in expressing information of patients in predicative phrases. However, most of these analyses depend on the intuitions of an individual researcher, which might not be shared by other people. In addition, their analyses mainly target single sentences, so the results cannot satisfy our need.

In our analysis of the non-referring NP components in a corpus of museum descriptions, we follow a more systematic approach. This is a part of the GNOME project (Generating NOMinal Expressions), which is a joint project between the University of Edinburgh and the University of Brighton. The project aims to build statistical models for NP form determination by training on annotated sub-corpora (Poesio, 2000a). A major result is a decision tree which assigns probabilities to different NP types according to the input semantic and discoursal features. The correctness of the decision trees relies heavily on the reliability of the annotation. As a result, GNOME emphasises on achieving reliable annotation, a stage which appears to have been skipped in much other work.

Our work on NP modifiers is along the same line. It aims at finding out reliable evidence for which information human authors like to convey in an NP and how they realise it. However, we do not intend to address the first problem in great detail because content selection is usually domain specific. Rules for one domain are not likely to be portable to a different domain since the communicative intentions and information needs can be very different. So we only intend to make strong claims about the second problem, i.e. given a piece of information, how it is realised in an NP. The relevant factors might include NP forms, the semantic properties of a piece of information, discourse properties and communicative intentions.

This paper is organised as follows: Section 2 describes the annotation of the features that we consider for each modifier; Section 3 analyses the results of corpus annotation, including the reliabilities of feature annotation and the regularities for modifier content selection and realisation; Section 4 describes the implementation of our observations in an NLG system called ILEX and a further evaluation of them using an unseen part of the annotated corpus; and finally in Section 5, we propose some future work.

2 Features of NP Modifiers

We have identified three features for each NP modifier (see (Poesio, 2000b) for features of NPs), which are:

- The pragmatic feature *pragm*: why is a modifier used in an NP;
- The semantic feature *sem*: which property of the entity denoted by an NP is expressed as a modifier in that NP;
- The realisation feature *type*: how is a given property expressed in an NP.

Through corpus annotation, we intend to answer the question of given the semantic and pragmatic features of an object property and some NP information, which syntactic forms (*type* values), along with probabilities, are likely to be used to realise the property. Note that it is not possible to answer questions like what is the chance of a semantic feature to be used for modification. The answer to this depends on the markup of the semantics of all the clauses in the corpus, whereas GNOME only considers NP features. Below we briefly introduce the major values of the three features.

Pragm We have observed three modifier functions in NPs. Firstly, providing properties to uniquely identify the objects or concepts denoted by an NP, i.e. components of the referring part of an NP. Such modifiers include most of those in generic references.

Secondly, having no effect in constraining a unique or unambiguous concept out of the NP which is either already unique or not required to have a unique interpretation, but being important to the situation presented in the main proposition containing the NP. This type is discussed in (Cheng and Mellish, 2000b) and it includes many modifiers in indefinite predicative phrases.

Thirdly, providing additional details about the referents of the NP, which functions the same way as the NP without these modifiers. The effect of such modifiers is usually local to the heads they describe rather than to the main propositions as a whole, which is the main difference between this and the second type of modifier.

The last two types form the non-referring part of an NP. In this paper, we mainly talk about the third type of modifiers.

Sem Based on what has been presented in the linguistic literature, e.g. (Levi, 1978; Quirk et al., 1985; Meyer, 1992), we summarise possible values of *sem*, some of which are listed in Table 1, where the numbers are the WordNet sense numbers for those particular concepts.

These values mainly target at prepositional phrases and noun modifiers, not adjectives. For an adjective, we use WordNet (Fellbaum, 1998) to derive its category rather than directly assigning a predefined value. In WordNet, if an adjective ascribes a value of a noun concept, e.g. *round* gives a value of the *shape* concept, WordNet will contain a pointer between the adjective and the noun by which the appropriate attribute is lexicalised, e.g. between *round* and *shape*. Using WordNet, annotating the *sem* feature of an adjective follows the steps of first choosing the correct sense for the adjective and then mapping the word sense to a noun concept in the WordNet ontology. Satisfactory agreement among human subjects on choosing senses for words has been recorded (Fellbaum, 1998) and the mapping to nouns can be done automatically. So this approach is considerably better than manually assigning values.

To avoid having too many categories, we use more general concepts like one of the followings to mark the *sem* feature of an adjective. For example, *round* is marked as `spatial-property1`.

`temporal-property1` : a property relating to time. For modifiers like *earlier, final.*

`visual-property1` : attributes of vision, including texture, lightness, colour, etc. For modifiers like *red, dark, superfine.*

`spatial-property1` : any property relating to or occupying space, including dimensionality, shape, form, etc. e.g. *round, hollow.*

`quality1` : an essential and distinguishing attribute of something or someone.

Type The values are given according to the syntactic characterisations of modifiers, including `appos` (appositive modifiers), `poss` (possessive determiners), `preadj` (adjectives before the head), `prenoun` (modifiers in the form of nouns or noun compounds before the head), `postprep` (prepositional phrases), `postpart` (present and past participles after the head noun), `postnp` (non-appositive modifiers in the form of noun phrases after the head noun), `rel-cls` (relative clauses).

`location1`	spatial positioning (at a point or in an area), for modifiers indicating where the object denoted by a head is located in physical space, including its origin. *a pattern of brass and pewter* ***on a tortoiseshell ground***
`time-period1`	temporal positioning (in a period of time), for modifiers that indicate the time period the object denoted by the head is located. *the French king Louis XIV's possessions* ***in 1720***
`material1`	cases where the modifier indicates the material of/from which the object denoted by the head is made. *This table's marquetry* ***of ivory and horn***
`identify2`	cases where the modifier names or identifies the referent of the head. The object denoted by the modifier is more specific than that denoted by the head. *the practice* ***of veneering furniture with marquetry of tortoiseshell, pewter and brass***
`rephrase1`	cases where the modifier "paraphrases" the lexical content of the head. In this case, the modifier and the head are equally specific and of the same type. *high blood pressure* ***(hypertension)***
`characterize1`	cases where the modifier provides general "characteristics" of the object denoted by the head. ***Finnish artist*** *Janna Syvanoja*
`subject7` or `object3`	cases in which the modifier occupies the subject or object role of the action denoted by the head. ***the boy's*** *application* (the boy applied for ...) ***the boy's*** *release* (... released the boy)
`possess` or `possinv`	possessive relations between the object(s) denoted by the head and the object(s) denoted by the NP in the modifier in a general sense. The relations include such subtypes as *whole/part, type/instance, owner/owned* etc. *desks* ***with interiors***, *the name* ***of the maker***

Table 1: Predefined semantic categories of modifiers

We wrote an annotation scheme manual for annotating NP modifiers, describing which elements of an NP should be marked as modifiers and how to mark their features. XML is used as the markup language. We used the same scheme to annotate referring and non-referring modifiers in all types of NPs, although we are more interested in the latter in referring expressions.

We trained two annotators to mark the NP modifiers in our corpus according to the manual and measured their agreement on a small part of the corpus. The annotated corpus contains 1863 modifiers, from which we could discover regularities in the usage of NP modifiers and design modifier generation algorithms based on these observations.

3 Results of Corpus Analysis

3.1 Agreement on Annotation

We use the kappa coefficient (K) measures (Siegel and Castellan, 1988) to calculate the agreement between annotators. According to (Carletta, 1996), a value of K between .8 and 1 indicates good agreement, and a value between .6 and .8 indicates some agreement. The agreement on the three modifier features by means of the Kappa statistic is:

Features	Type	Pragm	Sem
Agreement (K)	.97	.77	.81

This demonstrates fairly good agreement on *type* and *sem* and some agreement on *pragm.* For *sem*, there was a need to balance the number of values and the agreement because the more values a feature has, the less agreement the annotation can achieve. We use *other* for all infrequent properties and information expressed through relative clauses which cannot be classified to an existing category. About 10% of the *sem* features of the modifiers in our corpus are given the *other* value.

The agreement on *pragm* shows that the distinctions we are trying to make are relatively clear and human subjects can distinguish between the different uses of NP modifiers to some extent. The main ambiguity exists between `int` and `attr` modifiers. There seems to be a gradual difference between them and where to draw the line is a bit arbitrary. Some disagreement is also caused by the errors in the annotations of some NP features on which the annotation of *pragm* depends, although good agreements have been achieved on these features (Poesio, 2000b).

3.2 What is Expressed as a Modifier

Table 2 shows the distributions of the semantic and syntactic features of modifiers with respect to their functions[1]. It lists the number of modifiers found in the corpus for each *sem* and *pragm* combination and how many percentage such modifiers occupy in those with the same *pragm* value.

Comparing the columns, we find some differences in modifier usage. As to the *sem* property, `characterize1`, `spatial-property1`, `visual-property1` and `rephrase1` often give additional information. Among non-referring properties, `location1`, `identify2` and `material1` are more frequently expressed than others. `possess/possinv`, `quality1`, `temporal-property1`, `subject7` and `object3` tend to give referring information, and other categories have an almost equal chance of serving either function.

This observation gives the preferences for selecting additional properties to describe a discourse entity, that is, properties should be chosen according to such order: `characterize1`, `spatial-property1`, `visual-property1`, `rephrase1`, `location1`, `identify2` and `material1`, whereas the following properties should not normally be chosen: `possess/possinv`, `quality1`, `temporal-property1`, `subject7` and `object3`.

[1]The table only lists the main semantic and syntactic categories, so they do not necessarily add up to the amount in *Total*. In addition, `int` modifiers are not shown in the table.

SEM or **TYPE**	**PRAGM**	
	attr	unique
location1	88 (17.29%)	119 (12.13%)
possess or possinv	59 (11.59%)	239 (24.36%)
identify2	56 (11%)	111 (11.31%)
material1	45 (8.84%)	52 (5.3%)
time-period1	34 (6.68%)	58 (5.91%)
characterize1	33 (6.48%)	2
spatial-property1	32 (6.29%)	7
visual-property1	21 (4.13%)	18 (1.83%)
quality1	14 (2.75%)	76 (7.75%)
rephrase1	8 (1.57%)	0
subject7 or object3	7	77 (7.85%)
temporal-property1	3	32 (3.26%)
preadj	135 (26.52%)	337 (34.35%)
postprep	98 (19.25%)	293 (29.87%)
appos	77 (15.13%)	49 (4.99%)
postpart	66 (12.97%)	37 (3.77%)
prenoun	64 (12.57%)	105 (10.7%)
rel-cls	45 (8.84%)	18 (0.82%)
poss	2	124 (12.64%)
Total	509 (27%)	981 (52.66%)

Table 2: The distribution of Sem and Type with respect to Pragm

Linguistic work has shown that in human written texts, there are certain preferences about the place where new information usually appears. Fraurud observed from her corpus that 75% of the complex definite NPs were first mention (Fraurud, 1990), but her usage of complex NPs mainly refers to those with referring modifiers. In our corpus, we have found that a similar phenomenon holds for the use of non-referring information. 67.19% of such information appears in discourse new references, including bridging descriptions (21.41%), and only 11.79% in discourse old references. Therefore, additional properties of a discourse entity are usually given in the first mention of that entity.

3.3 How to Realise a Property

Table 2 also shows that referring and non-referring information can appear in any syntactic form. There is a tendency for `appos`, `postpart` and `rel-cls` to give additional information and postprep and poss to give referring properties.

In GNOME, we use the *wagon* CART *building program* developed at the Centre for Speech Technology Research, the University of Edinburgh to train a statistical model for deciding the syntactic form of a property given its semantic and pragmatic features and NP information. The construction of CART (Classification And Regression Trees) is a common and powerful method for building statistical models from simple feature data. The program has two parts: *wagon* and *wagon_test* which trains and tests a statistical model on given samples respectively.

Because the size of the annotated corpus is relatively small, we use a cross-validation method, which divides the corpus into two parts, 9/10s for training and 1/10 for testing. It runs the training and testing circle for 10 times and calculates overall correct prediction rate. The output of the program is a decision tree, whose intermediate nodes are questions concerning input features and leaf nodes probability density functions over all possible values of the predicated feature. A fragment of the decision tree trained from our corpus is given in Figure 1, where a leaf node specifies the choice of a realisation, the type with the largest probability, given all the conditions in the non-terminal nodes subsuming the leaf node.

The above process not only constructs a model for making realisation decisions, but also tests the accuracy of the model. At the moment, the global success rate for predicting a modifier realisation is 67.5%.

The accuracy rates with respect to *type* is given in Table 3. The table shows that the model performs well for appositive, possessive and adjectival modifiers, reasonably good for prepositional phrases and posthead NPs, but rather bad for prehead nouns and post-participles and the worst for relative clauses. The low accuracy is mainly the result of the less refined classification of semantic categories of modifiers. Properties realised as prenoun, postpart or rel-cls are very diverse and other is often used for simplification. Therefore, no specific semantic categories correlate with rel-cls and postpart. However, the low accuracy will not be a problem for generation because relative clause is a catch-all type of realisation and can be used to express all types of information.

Type	Total	Correct	Wrong	Percent
appos	124	110	14	88.70%
poss	154	136	18	88.31%
preadj	570	468	102	82.10%
prenoun	226	102	124	45.13%
postprep	402	288	114	71.64%
postpart	110	28	82	25.45%
postnp	38	24	14	63.15%
rel-cls	74	0	74	0%

Table 3: The accuracy rates with respect to *type*

pragm does not seem to play a big role in the decision tree for modifier realisation. This means that modifier usage does not determine or correlate with the choice of syntactic forms of modifiers.

The results given in this section have the same function as a set of embedding rules used in many NLG systems to generate NP modifiers. They can be transferred to embedding algorithms to make decisions concerning both content selection and realisation of NP modifiers.

4 Implementation in ILEX

We implemented the results of corpus analysis into the ILEX system, which adaptively generates online descriptions of museum objects. ILEX uses a modularised pipeline architecture, which consists of such modules as content selection, content structuring, text realisation, post-processing and presentation. The content of an NP is determined in the post-processing module by the NP generation and aggregation processes. This architecture allows us to substitute the original ILEX NP module with a new module which we call GNOME-edi[2].

GNOME-edi can generate several types of NP modifiers, including adjectives, prepositional phrases, relative clauses and appositive modi-

[2] A detailed description of GNOME-edi can be found in (Henschel and Cheng, 2000).

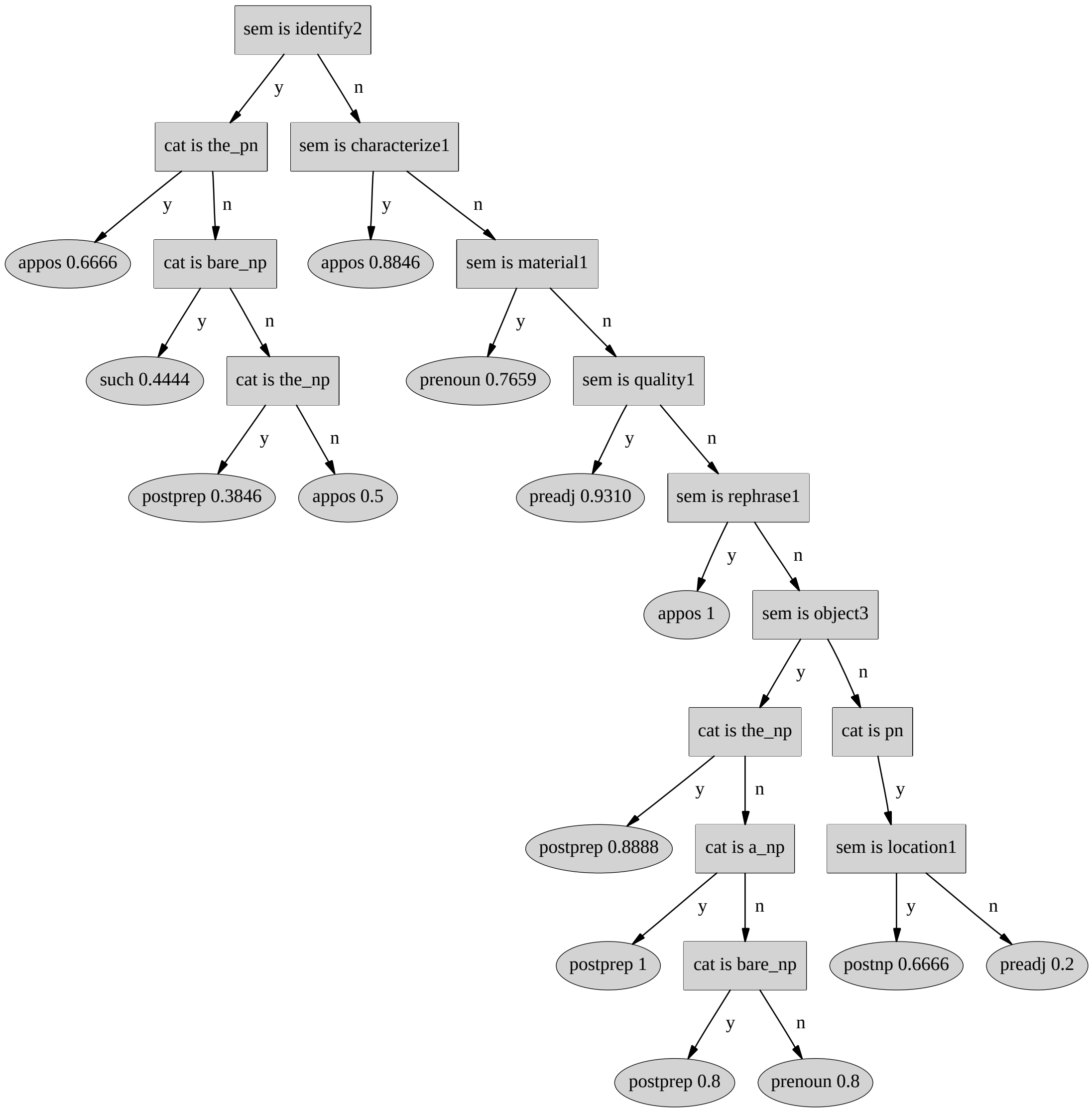

Figure 1: A fragment of the decision tree

fiers. It can generate disambiguation modifiers as well as modifiers providing additional information about a discourse entity.

The interface between `GNOME-edi` and the ILEX text realisation module contains an agenda, which consists of information decided by aggregation to be included into the NP that is being generated. Therefore the choice of NP forms takes into account aggregation considerations. `GNOME-edi` first chooses referring modifiers and then decides if more information can be expressed in the NP.

The selection of non-referring modifiers needs to consider such factors as NP types, the features of the current reference and the availability of syntactic slots around the NP head[3]. The generation of non-referring modifiers starts with the preferred properties (given in Section 3.2). If the discourse entity being realised has such a property, map it to a syntactic form preferred by the decision tree in Figure 1. This process continues until there is no more free slot around the head or preferred properties to be added.

Using `GNOME-edi`, ILEX is able to produce, in

[3]We assume fixed number of modifiers in an NP, e.g. three before head and two after.

a globally coherent text, sentences like: *This jewel is a necklace and was made by the Scottish designer Jessie King, who lived in London.* In this example, properties of `location1` (*Scottish* and *London*) and `identify2` (*Jessie King*) are chosen and an appositive construction is used to incorporate these information.

5 Conclusions and Future Work

This paper describes how we annotated the modifiers in a corpus of museum descriptions to discover rules for generating non-referring NP modifiers. We analysed the annotated corpus, which results in some rules about the selection and realisation of modifiers. We implemented the regularities in the ILEX system to generate complex NPs capable of serving multiple communicative intentions.

The work described in this paper can be improved in many ways, including more fine-grained analysis of the annotated corpus, increasing the size of the annotated corpus for training statistical models, which might achieve higher accuracy rate. The approach we have used allows us to test the model on a different domain as long as there exists annotated corpus for that domain. Actually we have annotated some instructions for using certain drugs. It is valuable to see how the model performs in a different domain.

References

Douglas Appelt. 1985. Planning english referring expressions. *Artificial Intelligence*, 26:1–33.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Hua Cheng and Chris Mellish. 2000a. Capturing the interaction between aggregation and text planning in two generation systems. In *Proceedings of INLG 2000*, Israel.

Hua Cheng and Chris Mellish. 2000b. An empirical analysis of constructing non-restrictive np modifiers to express semantic relations. In *Proceedings of INLG 2000*, Israel.

Hua Cheng. 1999. The annotation scheme for np heads and modifiers. Technical report, HCRC, Unit of Informatics, University of Edinburgh.

Robert Dale. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes.* The MIT Press.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database.* The MIT Press.

Kari Fraurud. 1990. Definiteness and the processing of noun phrases in natural discourse. *Journal of Semantics*, 7:395–433.

Barbara Grosz and Candace Sidner. 1986. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.

Renate Henschel and Hua Cheng. 2000. Gnome-edi document. Technical report, HCRC, the University of Edinburgh.

Helmut Horacek. 1995. More on generating referring expressions. In K. De Smedt, C. Mellish, and H. Novak, editors, *Proceeding of the 5th European Workshop on Natural Language Generation*, pages 43–58, Leiden, The Netherlands.

Judith Levi. 1978. *The Syntax and Semantics of Complex Nominals.* New York: Academic Press.

Charles Meyer. 1992. *Apposition in Contemporary English.* Cambridge University Press, Cambridge.

Massimo Poesio. 2000a. Annotating a corpus to develop and evaluate discourse entity realization algorithms: Issues and preliminary results. In *Proceedings of LREC 2000*, Greece.

Massimo Poesio. 2000b. The GNOME Annotation Scheme Manual. Url: www.cogsci.ed.ac.uk/ poesio/gnome/anno_manual_4.html, Centre for Cognitive Science, University of Edinburgh.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Grammar of Contemporary English.* Longman Group Ltd.

Jacques Robin. 1994. *Revision-based Generation of Natural Language Summaries Providing Historical Background: Corpus-based Analysis, Design, Implementation and Evaluation.* Ph.D. thesis, Computer Science Department, Columbia University.

James Shaw and Kathleen McKeown. 1997. An architecture for aggregation in text generation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Poster Session*, Japan.

Sidney Siegel and John Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. London:McGraw-Hill.