

# Corpus-based NP Modifier Generation

Hua Cheng and Massimo Poesio and Renate Henschel<sup>†</sup> and Chris Mellish

Division of Informatics, University of Edinburgh  
{huac,poesio}@cogsci.ed.ac.uk, chrism@dai.ed.ac.uk

<sup>†</sup>University of Bremen  
rhenschel@uni-bremen.de

## Abstract

This paper describes how we annotated and analysed the NP modifiers in a corpus of museum descriptions to discover rules for the selection and realisation of such modifiers, in particular non-referring ones. We implemented the regularities into an extension of the ILEX system to generate complex NPs capable of serving multiple communicative goals.

## 1 Introduction and Motivation

### 1.1 Generating Complex NPs

In addition to a referring function, noun phrases (NP) can also serve communicative goals such as providing new information about the referent and expressing the speaker's emotional attitude towards the referent (Appelt, 1985; Jordan, 2000). In Example (1) below, the part in italics refers to an object in a museum, and the part in boldface provides additional information about it.

(1) *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 humans do. 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. The italic part of (1) gives an example. 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, such as the part of (1) in boldface. If an NLG system generates such modifiers, it usually does this in its aggregation module, e.g. (Shaw and McKeown, 1997; O'Donnell et al., 1998).

This division is a functional one rather than a syntactic one. Except for the head and the determiner, which are always members of the referring part, other NP components can belong to different parts in different circumstances.

In NLG, research on NP generation focuses on deciding syntactic forms and choosing disambiguating modifiers for referring, e.g. (Dale, 1992). Other types of modifiers have been given much less attention, although proposals concerning their realisation are included in grammars such as NIGEL (Mann and Matthiessen, 1985). Work on aggregation is satisfied with devising a few rules to allow some degrees of embedding to generate subordinate NP components. There is no 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. Therefore, achieving coherence and conciseness are two conflicting considerations and conciseness should only be satisfied under coherence constraints.

## 1.2 Corpus Analysis for NLG

Corpus analysis has often been used for devising rules for modifier generation, e.g. (Mann and Matthiessen, 1985; Robin, 1994; Shaw and McKeown, 1997). The proposals in NIGEL, in particular, were the starting point of this work. 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 emphasise the conciseness consideration and might not suit multi-sentential text.

This paper puts emphasis on the generation of non-referring NP modifiers in museum descriptions, which provide information for items displayed in a museum. By analysing descriptions written and revised by human experts, we wish to find out general rules for the usage of such modifiers in the domain of museum descriptions (Oberlander et al., 1998), that hopefully will be general enough to be applied to other domains. Since what we will study are coherent human texts with multiple sentences, the NP modifiers in them are subject to both coherence and conciseness constraints.

This work is a part of the GNOME project (Generating NOMinal Expressions)<sup>1</sup>, in which statistical models for NP determination have been built by training on corpora annotated with semantic and discourse information (Poesio, 2000a). Such statistical models take the form of decision trees (Breiman et al., 1984), which assign probabilities to different NP types according to the input semantic and discursal features. The correctness of the decision trees relies heavily on the reliability of the annotation. As a result, GNOME emphasises achieving reliable annotation, a stage which appears to have been skipped in much other work.

In a similar way, our work aims at finding out reliable evidence concerning the information human authors like to convey as an NP modifier and how they realise it. However, we do not intend to address the content determination problem in great detail because content selection is usually domain specific. Our attention is in-

stead on 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 goals.

This paper is organised as follows: Section 2 describes the features of modifiers that we studied; Section 3 analyses the annotated corpus, 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; and finally in Section 5, we propose some future work.

## 2 Features of NP Modifiers

We have identified three main features for each NP modifier, 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*: which syntactic position is assigned to a given property in an NP? e.g. prenominal or postnominal, adjectival or as a relative clause.

Through corpus annotation, we wish to answer the question of what will be the probability of a given piece of information occupying a given syntactic position (a value of *type*) on the basis of the semantic and pragmatic properties of that information and relevant NP features, for example, whether a certain color attribute should be expressed by means of a prenominal adjective or a prepositional phrase in a definite NP. Notice that it is not possible to use corpus annotation to determine the likelihood of a given property to be chosen, unless we know in advance all of the properties that can be attributed to a given object, as in the case of Jordan's work on the COCONUT domain (Jordan, 2000). Below we briefly introduce the major values of the three modifier features.

**Pragm** We observed three modifier functions in NPs. Firstly, a modifier may specify properties that uniquely identify the objects or concepts denoted by an NP, i.e. components of the referring part of an NP. We call such modifiers

---

<sup>1</sup>GNOME is a joint project between the Universities of Brighton, Durham and Edinburgh. See (Poesio, 2000b) for features of NPs which are marked up in the GNOME corpus.

**uniq** modifiers; most modifiers in generic references are of this type.

Secondly, a modifier may not be used to constrain a unique or unambiguous concept out of an NP which is either already unique or not required to have a unique interpretation, but may be important to the situation presented in the main proposition containing the NP. We call this type **int** modifiers<sup>2</sup>; they include many modifiers in indefinite predicative phrases.

Finally, a modifier may be included to provide additional details about the referent of an NP which, however, would successfully function the same way as the NP without these modifiers. We call them **attr** modifiers. The effect of such modifiers is usually local to the heads they describe rather than to the main propositions as a whole. This is the main difference between **attr** and **int** modifiers.

The last two types of modifiers form the non-referring part of an NP. This paper emphasises **attr** modifiers and addresses **uniq** ones to show the differences between them.

**Sem** Our values of *sem* are based on the linguistic literature, e.g. (Levi, 1978; Quirk et al., 1985; Meyer, 1992). These values are also intended as a refinement of the semantic characterisations of the modifying relations in NIGEL (Mann and Matthiessen, 1985), where a correlation between certain semantic properties and the positions of modifiers is proposed. One of the modifications to this previous work was to use WordNet (Fellbaum, 1998) to make our values more standard and improve reliability. Some *sem* values are listed in Table 1, where the numbers are the WordNet sense numbers for these particular concepts.

These values mainly target prepositional phrases and nouns, not adjectives. For an adjective, we use WordNet to derive its category rather than directly assigning a predefined value. In WordNet, if an adjective ascribes a value to a noun concept, e.g. *round* gives a value of *shape*, 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 involves first choosing the correct sense for the adjective

---

<sup>2</sup>Some descriptions of **int** modifiers can be found in (Cheng and Mellish, 2000b).

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.

There is a tradeoff between the number of values and the achievable agreement on feature annotation because the more values a feature has, the less agreement the annotation can expect to achieve. To avoid having too many categories, we use more general concepts like one of the following (as defined in WordNet) to mark the *sem* feature of an adjective. For example, *round* in *the round table* would be marked as **spatial-property1**, which subsumes the *shape* concept.

**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 of this attribute specify the syntactic characterisations of modifiers, and include **appos** (appositive modifiers), **poss** (possessive determiners), **preadj** (adjectives before the head), **prenoun** (modifiers in the form of noun or noun compound before the head), **postprep** (prepositional phrases), **postpart** (present and past participles after the head), **postnp** (non-appositive modifiers in the form of noun phrase after the head) and **rel-clis** (relative clauses).

### 3 Results of Corpus Analysis

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 trained two annotators to mark the NP modifiers in the GNOME corpus according to the manual and measured

location1	spatial positioning, for modifiers indicating where the object denoted by a head is located in physical space, including its origin, e.g. <i>a pattern of brass and pewter on a tortoiseshell ground</i>
time-period1	temporal positioning, for modifiers that indicate the time period the object denoted by the head is located, e.g. <i>Louis XIV's possessions in 1720</i>
material1	cases where the modifier indicates the material of/from which the object denoted by the head is made, e.g. <i>This table's marquetry of ivory and horn</i>
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, e.g. <i>the practice of veneering furniture with marquetry of pewter and brass</i>
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, e.g. <i>high blood pressure (hypertension)</i>
characterize1	cases where the modifier provides general "characteristics" of the object denoted by the head, e.g. <i>Finnish artist Janna Syvanoja</i>
subject7 or object3	cases in which the modifier occupies the subject or object role of the action denoted by the head, e.g. <i>the boy's application</i> (the boy applied for ...), <i>the boy's release</i> (... 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 <i>whole/part</i> , <i>type/instance</i> , <i>owner/owned</i> etc., e.g. <i>desks with interiors, the name of the maker</i>

Table 1: Predefined semantic categories of modifiers

their agreement on a small part of the corpus. The annotated corpus contains 1863 modifiers. Using the annotated corpus, we could discover regularities in the usage of NP modifiers and design modifier generation algorithms.

### 3.1 Agreement on Annotation

We used the kappa coefficient (K) (Siegel and Castellan, 1988) to calculate the agreement between annotators. The agreement on the three modifier features is:

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

This demonstrates fairly good agreement on *type* and *sem* and some agreement on *pragm*. 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 introduced by errors in the annotations of the NP features on which the annotation of *pragm*

depends, e.g. logical form type and genericity<sup>3</sup>, although good agreements have been achieved on these features (Poesio, 2000a).

### 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<sup>4</sup>. Each cell gives the number of modifiers found in the corpus for each *sem/type* and *pragm* combination and what percentage such modifiers occupy in those with the same *pragm* value (vertically) and *sem/type* value (horizontally).

The percentages illustrate the differences in modifier usage. Concerning the semantic properties, **characterize1**, **spatial-property1**, **visual-property1** and **rephrase1** (percentages in boldface in the table) are more often given as additional information than as other types of information. Among the **attr** properties, some appear more frequently than others, for example, in decreasing frequencies these

<sup>3</sup> *Logical form type* specifies whether an NP is a quantifier, term or predicative, and *genericity* specifies whether the object denoted is a generic or specific reference.

<sup>4</sup> The table only lists the main semantic and syntactic categories, so they do not necessarily add up to the amount in *Total*. We do not include the categories with very low frequencies.

SEM or TYPE	PRAGM	
	attr(%,%)	uniq(%,%)
location1	88(17.3,38.3)	119(12.1,51.7)
possess/-inv	59(11.6, <u>16.4</u> )	239(24.4, <u>67.1</u> )
identify2	56(11,31.5)	111(11.3,62.4)
material1	45(8.8,40.9)	52(5.3,47.3)
time-period1	34(6.7,33.7)	58(5.9,57.4)
characterize1	33(6.5, <b>94.3</b> )	2(,5.7)
spatial-prop1	32(6.3, <b>69.6</b> )	7(,15.2)
visual-prop1	21(4.1, <b>41.2</b> )	18(1.8,29.4)
quality1	14(2.8, <u>8.4</u> )	76(7.8, <u>45.5</u> )
rephrase1	8(1.6, <b>100</b> )	0
subj7/obj3	7( <u>6.9</u> )	77(7.9, <u>75.5</u> )
temporal-prop1	3( <u>6.8</u> )	32(3.3, <u>72.7</u> )
preadj	135(26.5,20.5)	337(34.4,51.2)
postprep	98(19.3,19.4)	293(29.9,58)
appos	77(15.1, <b>60.2</b> )	49(5,38.3)
postpart	66(13, <b>54.1</b> )	37(3.8,30.3)
prenoun	64(12.6,32.7)	105(10.7,53.6)
rel-cls	45(8.8, <b>54.2</b> )	18(1.8,21.7)
poss	2	124(12.6,97.6)
<b>Total</b>	509(27.4)	981(52.7)

Table 2: The distribution of *sem* and *type* with respect to *pragm*

properties include: *location1*, *identify2*, *material1*, *time-period1*, *characterize1*, *spatial-property1* and *visual-property1*.

Properties such as *possess/possinv*, *location1*, *identify2*, *subject7/object3*, *quality1*, *time-period1*, *material1* and *temporal-property1* (in decreasing frequencies) tend to be used more often for referring. This gives a possible order for selecting properties to refer to a discourse entity. It seems to us that the semantic feature itself is far from sufficient for deciding the use of *int* modifiers, so we do not discuss them here.

Adding *attr* properties can cause confusion sometimes. For example, they might be read as referring information and confuse the reader about the referent. Avoiding such ambiguities in generation is important. So we rank *sem* values with significant preferences for serving the *attr* function, e.g. *characterize1*, over those occurring even more frequently as *attr* modifiers, e.g. *location1*. This suggests the following preferences for selecting *attr* properties to describe a discourse entity (A < B means A is preferred over B):

*rephrase1* < *characterize1* < *spatial-property1* < *visual-property1* < *location1* < *identify2* < *material1* < *time-period1*

And the following properties should not normally be chosen or should be down the list: *possess*, *possinv*, *quality1*, *subject7*, *object3* and *temporal-property1*.

The distribution of *type* values in Table 2 shows that a syntactic position can be used for any type of modifier. There is a tendency for appositive components, posthead participles and relative clauses (percentages highlighted in the table) to be used more often for realising *attr* properties, and possessive determiners, prehead adjectives and nouns and prepositional phrases more often for referring properties.

In addition, we have found a preference for non-referring information to appear in discourse new references (67.19%), including bridging descriptions (21.41%). Only 11.79% of such information appears in discourse old references and the rest is in predicative phrases.

### 3.3 How to Realise a Property?

To get a more precise correlation between a semantic and syntactic feature, we used 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. We also included the *cat* feature of the NP in which the property is realised, specifying its type (proper name, definite NP, pronoun, etc.) (Poesio, 2000b) as an approximation of the semantic and discourse features of the NP. 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 used 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 the average correct prediction rate. The output of the program is a decision tree, whose intermediate nodes are ques-

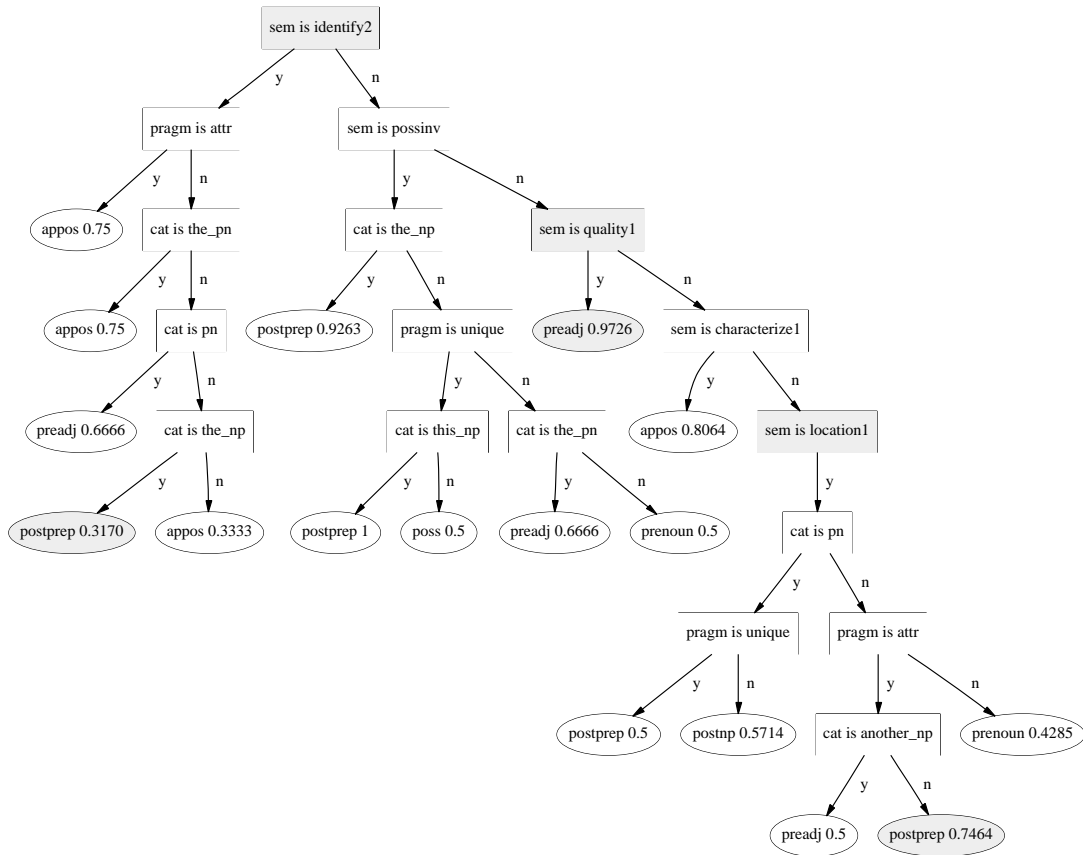


Figure 1: A fragment of the decision tree

tions concerning input features and leaf nodes probability density functions over all possible values of the predicted 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* value with the largest probability), given all the conditions in the non-terminal nodes subsuming the leaf node. Since a leaf node of the real tree contains in addition a list of type/probability pairs for all the values of *type*, it is possible to choose another syntactic position with equal or smaller probability to realise a property if so wish.

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 for museum descriptions is 67.5%. The accuracy rates with respect to the main values of *type* are shown in the third column of Table 3, where *Total* gives the number of modifiers of a specific type used for testing.

The rates show that the model performs well for appositive, possessive and adjectival modifiers, reasonably well for prepositional phrases and posthead NPs, but rather badly for prehead nouns and posthead participles and the worst for relative clauses.

The low accuracy is mainly the result of the less refined semantic classification. We used **other** for all infrequent properties which cannot be classified to an existing category. About 10% of the modifiers in our corpus are given this *sem* value. Properties realised as **prenoun**, **postpart** or **rel-cl**s are very diverse, so **other** is often used for simplification. There does not seem to be a correlation between specific semantic categories and **rel-cl**s or **postpart**. However, the low accuracy will not be a problem for NLG because relative clause is a catch-all type of realisation and can be used to express all types of information. For stylistic reasons, relative clauses are only used in NLG when other forms do not apply. Disregarding **rel-cl**s and

postpart, the global success rate would be 75%.

Type	Accuracy Rates			
	Museum Texts		Patient Leaflets	
	Total	Percent	Total	Percent
appos	124	88.70%	9	77.78%
poss	154	88.31%	55	98.18%
preadj	570	82.10%	59	96.61%
postprep	402	71.64%	44	63.64%
postnp	38	63.15%	2	0%
prenoun	226	45.13%	35	45.71%
postpart	110	25.45%	6	33.33%
rel-cls	74	0%	8	0%
<b>overall</b>	<b>1698</b>	<b>67.5%</b>	<b>218</b>	<b>75%</b>

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

The decision tree shows that *pragm* also plays a role in modifier realisation. This means that there are cases where modifier usage determines or correlates with the syntactic positioning of modifiers. However, it is difficult to single out its effect because of the complex dependencies between *sem*, *pragm* and *cat*.

In Section 4, we will show how the results discussed in this section can be employed in algorithms for making decisions concerning content selection and realisation of NP modifiers.

### 3.4 Domain Independence

The approach we have used allows us to test the realisation model on a different domain as long as there is an annotated corpus for that domain. We annotated a small corpus of patient information leaflets, which give instructions on how to use certain drugs. The average success rate of our decision tree in syntactic form prediction in this domain is 75% and its decomposition is given in the last column of Table 3. In general, our model is portable to this new domain. Some degradation in accuracy might be due to the small size of the test sample.

Often the same semantic property can be realised in many different ways, whereas Table 3 counts only one of them as correct. If instead realisations with equal probability are all treated as correct choices, the accuracy rate could be higher. However, we do not know whether such realisations are indeed favoured equally by humans, unless a separate experiment is carried out. So we did not consider such cases in calculating the accuracy. In addition, higher accuracy rate might also be achieved by training on

a larger annotated corpus.

## 4 Implementation in ILEX

We used the results discussed above in an extension of the ILEX system (Oberlander et al., 1998), 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.

GNOME-edi not only determines the form of an NP, but also generates several types of modifiers, including adjectives, prepositional phrases, relative clauses and appositive modifiers. 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 includes an agenda, which specifies the information that should be included in the NP to be generated according to the aggregation component. Therefore the choice of NP forms takes into account aggregation considerations. GNOME-edi first determines the NP type, then chooses referring modifiers (i.e. modifiers that function the same as those marked as *pragm* = **uniq** in the corpus) and finally decides if more information can be expressed in the NP (i.e. modifiers that function the same as those marked as *pragm* = **attr** in the corpus). In other words, the NP generation module decides a modifier function that is needed in an NP and then chooses a property and realisation that can fulfill this function.

The generation of non-referring modifiers needs to consider such factors as NP types, the semantic feature of a property and the availability of syntactic slots around the NP head<sup>5</sup>. The selection of non-referring properties starts with the preferred ones (given in Section 3.2). If the discourse entity being realised has such a property, it is mapped to a syntactic form preferred by the decision tree in Figure 1. This process

<sup>5</sup>We assume fixed number of modifiers in an NP, e.g. three before the head and two after.

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 a globally coherent text, sentences like: *This jewel is a necklace and was made by the important designer Jessie King from Scotland*. In this example, `identify2` (*Jessie King*) is a referring property and is realised as an appositive modifier (the second preferred realisation) because it does not have a PP form. Properties of `location1` (*Scotland*) and `quality1` (*important*) are chosen as non-referring modifiers and their realisations follow the decision tree (highlighted in Figure 1). The original ILEX NP module would however produce the phrase *an important Scottish designer called Jessie King*, where the realisation of the `identify2` property is not supported by our corpus analysis.

## 5 Conclusions and Future Work

This paper describes how we annotated and analysed the NP modifiers in a corpus of museum descriptions to discover rules for the selection and realisation of such modifiers, in particular non-referring ones. We implemented the regularities into an extension of the ILEX system to generate complex NPs capable of serving multiple communicative goals.

The work described here can be improved in many ways. A more fine-grained analysis of the annotated corpus is needed. We should also try to increase the size of the annotated corpus for training statistical models, which might help achieving higher accuracy rate.

## References

- D. Appelt. 1985. Planning English referring expressions. *Artificial Intelligence*, 26:1–33.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Trees*. Belmont, Calif.: Wadsworth International.
- H. Cheng and C. Mellish. 2000a. Capturing the interaction between aggregation and text planning in two generation systems. In *Proceedings of the 1st International Conference on Natural Language Generation*, pages 186–193, Israel.
- H. Cheng and C. Mellish. 2000b. An empirical analysis of constructing non-restrictive np components to express semantic relations. In *Proceedings of the 1st International Conference on Natural Language Generation*, Israel.
- R. Dale. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. The MIT Press.
- C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.
- B. Grosz and C. Sidner. 1986. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.
- P. Jordan. 2000. Can nominal expressions achieve multiple goals? an empirical study. In *Proceedings of the 38th Annual Meeting of Association for Computational Linguistics*, pages 142–149, Hong Kong, P.R.China.
- J. Levi. 1978. *The Syntax and Semantics of Complex Nominals*. New York: Academic Press.
- W. Mann and C. Matthiessen. 1985. Demonstration of the NIGEL text generation computer program. In James Benson and William Greaves, editors, *Systemic Perspectives on Discourse*, pages 50–83. Norwood: Ablex.
- C. Meyer. 1992. *Apposition in Contemporary English*. Cambridge University Press, Cambridge.
- J. Oberlander, M. O'Donnell, A. Knott, and C. Mellish. 1998. Conversation in the museum: Experiments in dynamic hypermedia with the intelligent labelling explorer. *New Review of Hypermedia and Multimedia*, 4:11–32.
- M. O'Donnell, H. Cheng, and J. Hitzeman. 1998. Integrating referring and informing in NP planning. In *Proceedings of COLING-ACL'98 Workshop on the Computational Treatment of Nominals*, pages 46–56, Montreal, Canada.
- M. Poesio. 2000a. Annotating a corpus to develop and evaluate discourse entity realization algorithms: Issues and preliminary results. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, pages 211–218, Athens, Greece.
- M. Poesio. 2000b. The GNOME annotation scheme manual. Technical Report URL: [www.cogsci.ed.ac.uk/~poesio/GNOME/anno\\_manual\\_4.html](http://www.cogsci.ed.ac.uk/~poesio/GNOME/anno_manual_4.html), Division of Informatics, the University of Edinburgh.
- R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Grammar of Contemporary English*. Longman Group Ltd.
- J. 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.
- J. Shaw and K. McKeown. 1997. An architecture for aggregation in text generation. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence, Poster Session*, Nagoya, Japan.
- S. Siegel and J. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. London:McGraw-Hill.