

XML Format Guidelines for the TUNA Corpus

Albert Gatt & Ielka van der Sluis & Kees van Deemter
Computing Science
University of Aberdeen

1 Introduction

This document forms part of the 2008 distribution of the TUNA Corpus, Version 1.0. This is the first public release of the complete TUNA Corpus of Referring Expressions. A subset of the corpus was used in the first Shared Task and Evaluation Challenge for NLG, the *Attribute Selection for the Generation of Referring Expressions Challenge* (ASGRE), co-located with the *Workshop on Using Corpora in NLG*. A subset is also being used for the second edition of the Challenge (the REG Challenge 2008), to be held in Ohio in June 2008, co-located with the *International Conference on NLG*. Both of these previous releases consist exclusively of the singular referring expressions in the TUNA corpus; moreover, the annotation for both ASGRE 2007 and REG 2008 has a different format which was specifically designed for the tasks involved.

This release contains the final version of the TUNA annotation, and includes the full corpus, that is, both singular and plural descriptions.

1.1 Version 1.0 distribution

The Version 1.0 corpus distribution contains the following:

README.txt	general README file to get you started
corpus/	the corpus of referring expressions
singular/	the singular descriptions in the corpus
people/	singular descriptions of people
furniture/	singular descriptions of furniture and household items
plural/	the plural descriptions in the corpus
people/	plural descriptions of people
furniture/	plural descriptions of furniture

	and household items
images/	the pictures used in the TUNA elicitation experiment
people/	the pictures for the people subcorpus
furniture/	the pictures for the furniture subcorpus
format.pdf	this document

2 Background to the TUNA Corpus

This document explains the XML format of the TUNA corpus data. This data is the result of a web-based experiment, in which participants were asked to describe objects (*targets*) in visual domains by typing and submitting referring expressions that distinguished them from other objects that were shown simultaneously (the *distractors*). Each experimental trial consisted of one (singular) or two (plural) targets, plus six distractors. The objects were shown in a sparse 3 (row) \times 5 (column) grid. In this way referring expressions were obtained of singular targets and plural targets (two objects) in two domains (people and furniture). For each trial in the experiment, the objects that were presented to the participants are known in terms of their type, properties and location, where the latter is defined in terms of their row and column. The TUNA Corpus is a semantically transparent corpus, in which the target(s) and its/their context are known.

The TUNA corpus was annotated with the main objective to evaluate the output of algorithms for the Generation of Referring Expressions (GRE) against the corpus data. GRE in principle involves everything from Content Determination to Realization, but here the focus is on Content Determination. GRE algorithms typically generate a list of properties with which the target referent can be described. These properties are commonly defined as attribute-value pairs. In order to be able to evaluate the algorithms' output, the annotation scheme concentrates primarily on attributes and values and logical structure (with a few extras to facilitate further research).

The realisation of referring expressions based on the selected properties was not a primary focus of TUNA. However, the annotation does allow the investigation of aspects of realisation, including the mapping from properties ('semantics') to word strings, should these be of interest.

Each file in the TUNA corpus data consists of a single *corpus instance*, which in turn represents an experimental *trial*. Each participant in the elicitation experiment carried out 38 trials, divided as shown in Table 1. Each corpus instance consists of one description by one participant (e.g. *the*

	SINGULAR	PLURAL
people	6	12
furniture	7	13
total	13	25

Table 1: Trials in the TUNA elicitation experiment

small red chair at the top, the man with the black beard) and the domain (description of entities and their attributes). The description is included in the corpus instance in three formats:

1. the description as written by the participant;
2. an annotated string that maps the description to substrings that represent attributes of the target;
3. an attribute set representation, which gives only the semantic content of the description

Section 3 of this document explains the files of the TUNA corpus and Section 4 gives a detailed definition of the XML format.

3 The TUNA Corpus Files and Directories

3.1 Filenames in the TUNA data

Each corpus instance is in a separate file. Filenames follow the naming convention `sNtM.xml`. Each filename is unique, and functions as a unique identifier for a corpus instance.

3.2 Images

The full set of images used in the TUNA elicitation experiment is also provided. Entities in the XML files have pointers to these images; see Section 4.2.1 below. The images can be found in the `images/` subdirectory in this distribution.

3.3 Object types

The TUNA data was elicited in two different domain types: There are references to photographs of people, and references to stylised pictures of furniture items (cf Table 1). The data is divided by domain type, with files in correspondingly named sub-directories. The images are also subdivided.

```

<TRIAL ID='sNtM'>

  <DOMAIN>
    representation of entities and their properties
  </DOMAIN>

  <STRING-DESCRIPTION>
    the string describing the target referent in the domain
  </STRING-DESCRIPTION>

  <DESCRIPTION>
    the string in STRING-DESCRIPTION, where the relevant substrings are annotated
    with attributes in ATTRIBUTE-SET
  </DESCRIPTION>

  <ATTRIBUTE-SET>
    the set of domain attributes which are true of the referent,
    and which are included in the description
  </ATTRIBUTE-SET>

</TRIAL>

```

Figure 1: Format of corpus instances

4 XML Format Description

This section gives a detailed description of the XML format of the TUNA corpus. First, a general overview is given of the XML structure of each file (i.e. corpus instance). More detailed definitions of the various components of this structure are given in the subsequent paragraphs of this section.

The basic format of corpus instances is shown in Figure 1. Each file contains an XML structure with a root node **TRIAL**. The definition of the **TRIAL** node is given in Section 4.1. A trial is structured such that it contains a **DOMAIN** node (See section 4.2), a **STRING-DESCRIPTION** node (See section 4.3), a **DESCRIPTION** node (See section 4.4) and an **ATTRIBUTE-SET** node (See Section 4.5).

4.1 The **TRIAL** node

The **TRIAL** node pairs domains and descriptions. This node has the following XML attributes:

1. **ID**: the unique corpus instance identifier. This is identical to the filename.

2. **CONDITION**: This takes one of two values: \pm LOC. The value of **CONDITION** is a reflection of the experimental condition in which the data was elicited. In the +LOC condition, participants were told that they could refer to entities using any of their properties, including their location. In the -LOC condition, they were *discouraged* from doing so, though not prevented. This means that, while it is more likely that descriptions in the +LOC condition contain locative expressions, this is not always the case. Moreover, there are some instances in the -LOC condition where participants did use locative expressions. The annotation of locatives is explained in Section 4.4. Note that \pm LOC is a between-subjects condition.
3. **CARDINALITY**: This specifies the number of referents (i.e., 1 or 2)
4. **SIMILARITY**: This takes one of two values \pm SIM. This attribute too is a reflection of the experimental condition in which a description was elicited; the **SIMILARITY** condition was within-subjects. On approximately half the plural trials (those with **CARDINALITY**=2), the two target referents had identical values on their distinguishing attributes (the +SIM condition); on the remaining plural trials, the two referents had different values on their distinguishing attributes. More specifically:
 - +SIM: The elements of the set have the same values on the attributes required to distinguish them. For example, if a description of the targets needed the attribute `colour` to be distinguishing, both targets might have the value `blue` for this attribute.
 - -SIM: The elements of the set have different values on the attributes required to distinguish them. For example, if a description in the `people` domain required the attribute `hasGlasses`, then one target referent might have glasses, while the other doesn't. A description could then be *the man with glasses and the man without glasses*.

NB: All singular trials take the value +SIM by default.

5. **DOMAIN**: This attribute specifies the object type in a **TRIAL**, i.e. `furniture` or `people`.

4.2 The DOMAIN node

The DOMAIN node represents the entities in a domain. The domain consists of one or two target referents (depending on whether CARDINALITY is 1 or 2) and six distractors. Each entity is represented as a separate sub-node of the DOMAIN node, called ENTITY. The overall structure looks like this:

```
<DOMAIN>
  <ENTITY>
    <ATTRIBUTE />
    <ATTRIBUTE />
    ...
  </ENTITY>
  ...
</DOMAIN>
```

The ENTITY and the ATTRIBUTE node are defined in the following two subsections.

4.2.1 The ENTITY node

An ENTITY node consists of a set of attributes, corresponding to a description of a domain object. The ENTITY node has the following XML attributes:

1. **ID**: A unique integer identifier.
2. **IMAGE**: The filename containing the picture of this entity. These images are found in the `images/` directory provided with this distribution.
3. **TYPE**: This specifies whether an entity in this specific corpus instance was a *target* or a *distractor*. There can be one or two targets, depending on whether the TRIAL is singular or plural. There are always six distractors. The target(s) is/are the intended referent(s) which the description is intended to identify, relative to the domain distractors.

4.2.2 ATTRIBUTE nodes

Every ENTITY node has a number of child ATTRIBUTE nodes, representing properties of the entity in attribute-value notation. The set of attributes and possible values in each domain (furniture/people) is displayed in Table 2. Each has the following XML attributes:

- **NAME**: the name of the attribute e.g. `colour` or `size`

Attributes in the furniture/household domain		
Attribute	Type	Possible values
TYPE	literal	<i>chair, sofa, desk, fan</i>
COLOUR	literal	<i>blue, red, green, grey</i>
SIZE	literal	<i>large, small</i>
ORIENTATION	literal	<i>left, right, front, back</i>
X-DIMENSION (column number)	gradable	1, 2, 3, 4, 5
Y-DIMENSION (row number)	gradable	1, 2, 3
OTHER		see Section 4.4.3
Attributes in the people domain		
Attribute	Type	Possible values
TYPE	literal	<i>person</i>
ORIENTATION	literal	<i>front, left, right</i>
AGE	literal	<i>young, old</i>
HAIRCOLOUR	literal	<i>dark, light, other</i>
HASBEARD	literal	<i>yes, no, dark, light, other</i>
HASHAIR	literal	<i>yes, no, dark, light, other</i>
HASGLASSES	boolean	0 (false), 1 (true)
HASSHIRT	boolean	0 (false), 1 (true)
HASTIE	boolean	0 (false), 1 (true)
HASSUIT	boolean	0 (false), 1 (true)
X-DIMENSION (column number)	gradable	1, 2, 3, 4, 5
Y-DIMENSION (row number)	gradable	1, 2, 3
OTHER		see Section 4.4.3

Table 2: Attributes and values used in the TUNA corpus

- **VALUE**: the value of the attribute e.g. *red* or *large*
- **TYPE**: the value type. This is either *literal* (where the value is an element of a set of possible values), *boolean* (the value is 1 or 0) or *gradable*, meaning the attribute takes a numeric value. There are only 2 gradable attributes, *x-dimension* and *y-dimension*, corresponding to the column (X) and row (Y) of the entity in the domain grid.

An example of an ENTITY node, corresponding to the picture in Figure 2, is shown below:

```
<ENTITY ID="80" IMAGE="chairRightRedSmall.gif" TYPE="target">
```



Figure 2: One of the stimuli in the TUNA elicitation experiment

```
<ATTRIBUTE NAME="colour" TYPE="literal" VALUE="red"/>  
<ATTRIBUTE NAME="orientation" TYPE="literal" VALUE="right"/>  
<ATTRIBUTE NAME="type" TYPE="literal" VALUE="chair"/>  
<ATTRIBUTE NAME="size" TYPE="literal" VALUE="small"/>  
<ATTRIBUTE NAME="x-dimension" TYPE="gradable" VALUE="1"/>  
<ATTRIBUTE NAME="y-dimension" TYPE="gradable" VALUE="3"/>  
</ENTITY>
```

4.3 The STRING-DESCRIPTION node

This is the description of the target entity, as written by the human participant, provided without any annotation. Some of these descriptions may contain punctuation, and use either upper or lower-case letters. An example is shown below:

```
<STRING-DESCRIPTION>  
  A red chair which is facing the  
  left hand side of the screen  
</STRING-DESCRIPTION>
```

4.4 The DESCRIPTION node

This consists of the string in the STRING-DESCRIPTION node, where each substring is annotated with the attribute it represents. The idea is to map substrings to the relevant parts of the target entity's domain representation. The DESCRIPTION also marks up the main determiner, if any. The DESCRIPTION node has the following attribute:

- **NUM**: This takes the value **SINGULAR** or **PLURAL**. Singular expressions are descriptions that describe one object. In the TUNA Corpus this tag was used in two situations:
 1. If the cardinality of the reference set as specified in the description equals 1, the complete subject input was marked as singular;
 2. In plural descriptions in which two independent object descriptions are combined (i.e., "the fan and the sofa"), each of the two object descriptions was marked as singular (see Section 4.4.5 for more on these cases). Plural expressions are descriptions describing two objects.

DESCRIPTION nodes can be nested; this only happens in the case of plurals, where the nesting of DESCRIPTIONs indicates the logical form of the description. In addition, DESCRIPTION nodes contain zero or more ATTRIBUTE nodes, which enclose substrings and identify the domain attribute that they represent (but see Section 4.4.3). A special kind of attribute tag, called META-ATTRIBUTE can also be included. An example of a plural description is shown in Figure 3. The rest of this section explains its various aspects in detail.

4.4.1 ATTRIBUTE and META-ATTRIBUTE nodes

A DESCRIPTION node may have one or more ATTRIBUTES as children; these are the target referent's properties, as specified in the DOMAIN, which a human author included in his/her description. ATTRIBUTE nodes are annotated identically to the corresponding ATTRIBUTES in the DOMAIN, except that they are assigned an arbitrary ID, indicating that the ATTRIBUTE in a DESCRIPTION is a realisation of the ATTRIBUTE in the DOMAIN.

For locative expressions, the DESCRIPTION uses a META-ATTRIBUTE. This encloses the relevant expression (e.g. the string *in the middle row* in Figure 3), and has the following attributes:

- **NAME**: This is the name of the attribute, and is always **location**.
- **VALUE**: This can have the following values to denote the general spatial information included in the description: **left**, **right**, **top**, **bottom**, **middle** and **other**, where:
 - The values **left**, **right**, **top**, and **bottom** are only used if the locative expression explicitly uses these orientations. For exam-

```

<DESCRIPTION NUM="PLURAL">
  <DET ID="27" VALUE="definite">the </DET>
  <DESCRIPTION NUM="PLURAL">
    <DESCRIPTION NUM="SINGULAR">
      <META-ATTRIBUTE ID="a151" NAME="location" VALUE="right">
        <ATTRIBUTE ID="a1" NAME="x-dimension" VALUE="4"/>
        rightmost
      </META-ATTRIBUTE>
    </DESCRIPTION>
    and
    <DESCRIPTION NUM="SINGULAR">
      <META-ATTRIBUTE ID="a153" NAME="location" VALUE="left">
        <ATTRIBUTE ID="a2" NAME="x-dimension" VALUE="2"/>
        left most
      </META-ATTRIBUTE>
    </DESCRIPTION>
  </DESCRIPTION>
  <ATTRIBUTE ID="a3" NAME="type" VALUE="other">pictures </ATTRIBUTE>
  <META-ATTRIBUTE ID="a156" NAME="location" VALUE="middle">
    <ATTRIBUTE ID="a4" NAME="y-dimension" VALUE="2"/>
    in the middle row
  </META-ATTRIBUTE>
</DESCRIPTION>

```

Figure 3: Example of a DESCRIPTION node

ple, the expression *rightmost* in Figure 3 is marked up as having `VALUE="right"`.

- The value `middle` is only used if the locative expression explicitly refers to the middle. For example, the expression *in the middle row* in the Figure is marked up as having the value `middle`. However, `middle` can mean either (a) middle column; (b) middle row; or (c) absolute middle of the screen, depending on what the author of the description meant. Thus, the expression *in the middle row* in Figure 3 refers to the row.
- The value `other` is used if none of the above applies.
- **REL**: This is an optional attribute, and is present only if the locative expression is relational. For example, the expression *to the left of the red chair* would have a **REL** attribute, whose value is the integer ID of the **ENTITY** which the locative refers to. For example:

```
<META-ATTRIBUTE ID="..." NAME="location" REL="122">  
  <ATTRIBUTE ID="..." NAME="x-dimension" VALUE="4"/>  
  next to the green fan  
</META-ATTRIBUTE>
```

contains a reference to the **ENTITY** with ID 122 in the **DOMAIN**.

The **META-ATTRIBUTE** has, as its child nodes, either an **x-dimension** attribute, a **y-dimension** attribute or both. It has both just in case the locative expression is a combination of both the vertical (Y) and the horizontal (X) position of the target referent (usually when it is an expression referring to the absolute middle of the grid). Note that:

- The **x-dimension** **ATTRIBUTE** is assigned the number of the column in which the target was presented in the trial. Its presence indicates that the locative expression refers to the horizontal location of the target referent (column number). In Figure 3 the **META-ATTRIBUTES** enclosing *leftmost* and *rightmost* have this **ATTRIBUTE** as daughter.
- The **y-dimension** attribute is assigned the number of the column in which the target was presented in the trial. Its presence indicates that the locative expression refers to the horizontal location of the target referent (column number). In Figure 3 the **META-ATTRIBUTES** enclosing *in the middle row* has this **ATTRIBUTE** as daughter.

The reason why the `META-ATTRIBUTE` was included, is that a single locative expression does not correspond to a single location (X or Y) value in one-to-one fashion. For example, the expression *in the centre of the screen*, where the centre is a function of both row and column, is annotated as follows:

```
<META-ATTRIBUTE ID="..." NAME="location">
  <ATTRIBUTE ID="..." NAME="x-dimension" VALUE="3" />
  <ATTRIBUTE ID="..." NAME="y-dimension" VALUE="2" />
  in the centre of the screen
</META-ATTRIBUTE>
```

4.4.2 Nesting of `ATTRIBUTE` nodes

`ATTRIBUTE` nodes can sometimes be nested. This happens in one, and only one, case, namely, where there is an `ATTRIBUTE` whose `NAME="hairColour"`. This is the daughter of another `ATTRIBUTE` node, because the use of `hairColour` always entails that either `hasHair` or `hasBeard` has the value 1. For example, the expression *white-haired* is annotated as follows:

```
<ATTRIBUTE ID="..." NAME="hasHair" VALUE="1">
  <ATTRIBUTE ID="..." NAME="hairColour" VALUE="light" />
  white-
</ATTRIBUTE>
  haired
</ATTRIBUTE>
```

4.4.3 `ATTRIBUTE` nodes with `NAME="other"`

Occasionally, a description referred to properties of an object that had not been included in the domain representation. This is more frequent in the `people` than in the `furniture` domain, and is found in expressions such as *with the moustache*, which would be annotated as follows:

```
<ATTRIBUTE ID="..." NAME="other" VALUE="other">
  with the moustache
</ATTRIBUTE>
```

4.4.4 `ATTRIBUTE` nodes with `VALUE="other"`

Sometimes, human-authored descriptions contain properties which are clearly values to one of the domain `ATTRIBUTE`s of an `ENTITY`, but do not correspond

	syntactic rule	semantic rule
1.	$D_{sg} \rightarrow A_1, \dots, A_n$	$\llbracket D_{sg} \rrbracket = A_1 \wedge \dots \wedge A_n$
2.	$D_{pl} \rightarrow A_1, \dots, A_n$	$\llbracket D_{pl} \rrbracket = A_1 \wedge \dots \wedge A_n$
3.	$D_{pl} \rightarrow D_1, \dots, D_n$	$\llbracket D_{pl} \rrbracket = \llbracket D_1 \rrbracket \vee \dots \vee \llbracket D_n \rrbracket$
4.	$D_{pl} \rightarrow D, A$	$\llbracket D_{pl} \rrbracket = \llbracket D \rrbracket \wedge A$

Figure 4: Rules for the interpretation of descriptions using the XML data

to the value listed in the DOMAIN for that ENTITY. For example, Figure 3 contains the substring *picture*. This is clearly the ‘type’ of the entities in question (i.e. it is the way the entities have been categorised by the author). Hence, the substring is annotated as an ATTRIBUTE whose name="type", but whose value="other".

4.4.5 Plural Descriptions and nesting

In Figure 3, the whole is enclosed in a plural DESCRIPTION tag, indicated by the NUM attribute. In addition to a DET, the description itself is composed of a number of ATTRIBUTE and META-ATTRIBUTE nodes, with a further daughter plural DESCRIPTION tag. The latter then encloses two singular DESCRIPTIONs, each of which is a coordinate in an NP. This permits the compilation of the logical form of the description, and also resolves ambiguities. For example, in the Figure, the expression *in the middle row* could either apply to the object denoted by *leftmost* or to both objects denoted by the coordinate NP *rightmost and leftmost*. These ambiguities were resolved by referring to the DOMAIN representation (in this case, both target referents are in the middle row, hence it’s the second interpretation). The expression is disambiguated by having the location META-ATTRIBUTE modify the nested plural DESCRIPTION.

On the basis of this simple syntactic markup, a logical form can be derived compositionally. The derivation of a logical form is achieved by the recursive application of the semantic rules shown in Figure 4. The left hand side of the figure shows syntactic rules, in a context-free grammar format. Rules 1 and 2 stipulate that a singular or plural DESCRIPTION tag (denoted D_{sg} and D_{pl}) could have any number of ATTRIBUTE or META-ATTRIBUTE tags (A) as children. The corresponding semantic form is a conjunction. A plural description can also be composed of several embedded descriptions (rule 3). Description nodes which are siblings in the XML tree are disjoined. On the other hand, a description whose sibling is an attribute node is conjoined to the semantic representation of that node (rule 4). Using these rules, the

example description in the Figure yields the following logical form:

$$\begin{array}{ccc}
 \langle X\text{-DIMENSION} : 4 \rangle & \vee & \langle X\text{-DIMENSION} : 2 \rangle \\
 & \wedge & \\
 \langle \text{TYPE} : \textit{picture} \rangle & \wedge & \langle Y\text{-DIMENSION} : 2 \rangle
 \end{array}$$

4.5 The ATTRIBUTE-SET node

The ATTRIBUTE-SET is the set of ATTRIBUTE nodes that the DESCRIPTION contains. It represents a ‘flat’ semantics, i.e. the set of attribute-value pairs included in a description, without reference to their syntactic structure or logical form. For example, the ATTRIBUTE-SET corresponding to Figure 3 is as follows:

```

<ATTRIBUTE-SET>
  <ATTRIBUTE ID="a1" NAME="x-dimension" VALUE="4"/>
  <ATTRIBUTE ID="a2" NAME="x-dimension" VALUE="2"/>
  <ATTRIBUTE ID="a3" NAME="type" VALUE="other"/>
  <ATTRIBUTE ID="a4" NAME="y-dimension" VALUE="3"/>
</ATTRIBUTE-SET>

```

Note that the ID value for ATTRIBUTE nodes in the DESCRIPTION and ATTRIBUTE-SET is identical. META-ATTRIBUTE nodes are not included in this representation.

Occasionally, an ATTRIBUTE-SET may contain the same ATTRIBUTE twice (i.e. two ATTRIBUTE nodes with the same NAME and VALUE, but different IDs). This usually occurs with plural descriptions. For example, the description *the blue chair and the blue desk* has two occurrences of the word *blue*, which maps to the property $\langle \text{COLOUR} : \textit{blue} \rangle$.

5 Further information

The publications listed below report studies made on the corpus, as well as details of the design of the TUNA elicitation experiment, annotation procedures, and inter-annotator agreement. All the papers are available from the TUNA Project webpage¹. A journal paper on the corpus as a whole (focusing on singulars) is in preparation.

¹<http://www.csd.abdn.ac.uk/research/tuna/>

van Deemter, K., van der Sluis, I., and Gatt, A. (2006). Building a semantically transparent corpus for the generation of referring expressions. *Proceedings of the 4th International Conference on Natural Language Generation (INLG), (Special Session on Data Sharing and Evaluation)* [A preliminary description of the corpus, its design and its purpose]

Gatt, A., van der Sluis, I., and van Deemter, K. (2007) Evaluating algorithms for the generation of referring expressions using a balanced corpus. *Proceedings of the 11th European Workshop on Natural Language Generation, ENLG-07* [A description of the elicitation experiment, annotation, and an evaluation of some GRE algorithms on the furniture sub-corpus]

van der Sluis, I., Gatt, A., and van Deemter, K. (2007). Evaluating Algorithms for the Generation of Referring Expressions: Going Beyond Toy Domains. *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP-07* [A further evaluation of classic GRE algorithms on the people sub-corpus]

6 Acknowledgements

The work reported in this document formed part of the TUNA Project, supported by EPSRC Grant no. GR/S13330/01. We are grateful to Imtiaz Hussain Khan and Ross Turner for assistance with the annotation. We would like to thank Richard Power, Emiel Krahmer, and Anja Belz for helpful comments.