Generation of Dutch referring expressions using the D-TUNA corpus

Marissa Hoek (m.d.hoek@student.utwente.nl)
Human Media Interaction, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands

Mariët Theune (m.theune@utwente.nl)
Human Media Interaction. University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

This paper describes our research into generating Dutch noun phrases as descriptions of furniture objects or people. This is usually done in two steps: attribute selection and realisation. This research focuses only on the realisation step: generating a noun phrase from given attributes. The research is done on the Dutch version of the TUNA-corpus, which contains annotated human-produced descriptions.

Three algorithms were developed for this task, each an improvement over the last. We extracted the lexical choice from the D-TUNA corpus, and used templates generated from the corpus which specified the order of attributes. The algorithms were then evaluated for string similarity with the original human descriptions from the corpus, and a human evaluation was carried out which tested clarity and fluency. A steady improvement of scores in both the automatic and human evaluation was observed for each new version of the algorithm.

Keywords: Referring expression generation; Dutch; realisation; evaluation; corpus analysis

Introduction

The automatic generation of object descriptions in natural language is an important research topic in the field of natural language generation (NLG). Given a number of objects in a visual scene, the task of referring expression generation is to create a description that allows the human user to identify the intended target object. The task is usually carried out in two steps:

1. Attribute selection: select a set of properties that uniquely characterize the target object (and none of the other objects in the scene)

2. Realisation: create a noun phrase in natural language that expresses these properties, for example "the dark-haired man with the glasses".

This research focuses on the realisation step: creating a Dutch noun phrase that expresses the properties selected by the attribute selection algorithm by using information from a corpus.

Brugman et al. (2009) developed a template-based realisation method for the TUNA-STEC (Gatt et al., 2009). Their templates were manually generated. It would be more efficient to automatically generate templates. Di Fabbrizio et al. (2008) describe a surface realisation strategy that does this, by taking an annotated string from the corpus and replacing text segments that represent a property with the corresponding attribute type. Hervás et al. (2013) describe a case-based reasoning approach to the realisation of referring expressions. They extract lexical choice from the English TUNA corpus, taking personal preferences into account. Their results show that the generated referring expressions are more similar to the original human expressions compared to systems that do not take personal preference into account. Other approaches to the realisation of referring expressions have employed grammars (Gervás et al., 2008), bigram language models (Pereira and Paraboni, 2008) and graph transducers (Bohnet, 2008).

Here we present a similar approach to Di Fabbrizio et al. (2008), but for Dutch. We also investigate the effect of corpus-based lexicalization. To our knowledge, this is the first research into corpus-based realization of Dutch referring expressions.

The paper starts by describing the D-TUNA corpus we used for this research. After that we discuss the development process by describing the three algorithms and the techniques used to gradually improve the system. Then we describe the evaluation process and describe and discuss the results of this evaluation. In the final section we conclude on our findings.

Corpus description

For this research we used the D-TUNA corpus (Koolen and Krahmer, 2010). This corpus is a Dutch version of the TUNA corpus (van Deemter et al., 2006). It contains 2400 Dutch descriptions. The descriptions were produced by test subjects after being shown a visual scene with distractor and target objects, as can be seen in Figure 1. The goal was to provide distinguishing descriptions of the target objects. The test subjects were instructed to only use inherent properties of the target objects; they were not allowed to use the location. The descriptions are of furniture and people, both singular and multiple objects or persons. The corpus contains written descriptions, transcribed spoken descriptions and transcribed descriptions from face-to-face conversations. Each category contains 200 descriptions, resulting in a total of 2x2x3x200 = 2400 descriptions. For this research we only use the singular written descriptions of furniture and people. The portion of the corpus we used was split into three parts: a training set of 120 trials per domain, and two test sets of 40 trials each. Of these two test sets, we only used one.

Each trial contains, in XML-format, the properties of the target object and distractor objects in the scene. It also contains the description that was written by the test subject, in

both raw and annotated form. Each word in the description that describes an attribute of the target object is annotated with its type and value. An example of an annotated description is shown in Figure 2. The different attributes and the possible values for each domain are shown in Tables 1 and 2.

Figure 1: A sample scene, as was shown to the test subjects (Koolen and Krahmer, 2010).

```
<DESCRIPTION NUM="singular">
 <ATTRIBUTE ID="a1" NAME="type" VALUE="person">meneer</ATTRIBUTE>
 <ATTRIBUTE ID="a2" NAME="hasBeard" VALUE="1">met een baard</ATTRIBUTE>,
 <ATTRIBUTE ID="a3" NAME="hasGlasses" VALUE="1">een bril</ATTRIBUTE> en
 <ATTRIBUTE ID="a5" NAME="hasHair" VALUE="1"><ATTRIBUTE ID="a4" NAME=
"hairColour" VALUE="dark">vrij donker</ATTRIBUTE> haar</ATTRIBUTE>
</DESCRIPTION>
```

Figure 2: A sample XML-description from the D-TUNA corpus.

Algorithms

This section describes the development process of the three realization algorithms.

Algorithm 1: Baseline

The first algorithm is a basic algorithm for generating descriptions, which works by filling in fixed templates, one for the people domain and one for the furniture domain. The templates are as follows:

Furniture:
"{det}{size}{colour}{type}{orientation}"

People:
"de {age}{hairColour}man{has}{hasNot}{orientation}"

Where {det} represents the article, and {size}, {colour}, {type}, {orientation}, {age} and {(hairColour} represent attributes. {has} and {hasNot} represent lists of properties that are present in the target object ("met een", with a) or absent ("zonder", without). These properties are added in the order hasBeard, hasTie, hasHair, hasSuit, hasGlasses, hasShirt.

Attribute	Possible values
Type	Chair, sofa, desk, fan
Colour	Blue, red, green, grey
Orientation	Front, back, left, right
Size	Large, small

Table 1: The attributes and values in the furniture domain.

Attribute	Possible values
Type	Person
Orientation	Front, left, right
Age	Young, old
Hair colour	Dark, light, other
Has hair	0 (false), 1 (true)
Has beard	0,1
Has glasses	0,1
Has shirt	0,1
Has tie	0,1
Has suit	0,1

Table 2: The attributes and values in the people domain.

Dutch has two definite articles. The algorithm uses a function with a little dictionary, which returns the correct article for each noun. There is also a function for inflecting adjectives (e.g. "rood" (red) becomes "rode" when preceded by a definite determiner), which also uses a little dictionary. The words and phrases used in this algorithm are straightforward translations of the English attribute values from the corpus.

In order to create correct noun phrases, we have to know something about the ordering of words and modifiers in a sentence. The Algemene Nederlandse Spraakkunst (Geerts et al., 1984), a book on the structure of Dutch language, has a chapter about the structure of noun phrases (ch.14). This chapter mentions the order of adjective modifiers for nouns. It states that adjectives having to do with the material of the noun (type C) come last. To the left of that are adjectives having to do with shape or colour (type B), and other adjectives are even to the left of that (type A). The order is thus *other - shape/colour - material - noun*, which corresponds to the order *size - colour - type* for the furniture domain.

Most other modifiers, such as the orientation in the furniture domain and most attributes in the people domain, can be expressed with prepositional phrases. According to ANS chapter 14.6.2, these come after the noun. The ANS does not mention a specific order for when multiple prepositional phrases are used, like when we want to express that a person has a beard and glasses. This does not mean there is no specific order that is 'better' or that feels more natural for a certain combination of features. In our final algorithm, we try to extract such orders from the corpus.

Attribute	Baseline	Translation	Improved Lexicalization	Frequency
Type.sofa	bank	sofa	bank	100%
Type.fan	ventilator	fan	ventilator	77%
Type.desk	bureau	desk	bureau	83 %
Type.chair	stoel	chair	stoel	100%
Orient.front	gezien vanaf voren	seen from the front	van de voorkant gezien	*
Orient.back	gezien vanaf achteren	seen from the back	van de achterkant gezien	*
Orient.left	gezien vanaf links	seen from the left	die naar links is gedraaid	*
Orient.right	gezien vanaf rechts	seen from the right	die naar rechts is gedraaid	*
Size.small	kleine	small	kleine	76%
Size.large	grote	large	grote	84%
Color.red	rode	red	rode	100%
Color.blue	blauwe	blue	blauwe	100%
Color.grey	grijze	grey	grijze	95%
Color.green	groene	green	groene	100%

Table 3: The phrases used in the furniture set. An asterisk denotes an exception from the rule of choosing the most frequent lexicalization. This is further explained in the text.

Attribute	Baseline	Translation	Improved Lexicalization	Frequency
Hair.light	lichtharige	light-haired	grijs haar	32%
Hair.dark	donkerharige	dark-haired	donker haar	56%
Age.young	jonge	young	jongere	50%
Age.old	oude	old	ouder uitziende	100%
hasTie	stropdas	tie	stropdas	94%
hasHair.0	zonder haar	with no hair	kaal hoofd	40%
hasShirt	shirt	shirt	witte blouse	63%
hasSuit	pak	suit	pak	54%
hasGlasses	bril	glasses	bril	94%
hasBeard	baard	beard	baard	91%
Orient.left	gezien vanaf links	seen from the left	die naar links kijkt	60%
Orient.right	gezien vanaf rechts	seen from the right	die naar rechts kijkt	67%
Type.man	man	man	man	97%

Table 4: The phrases used in the people set.

Algorithm 2: Improved lexicalization

The main improvement for the 'improved lex' algorithm lies in the way it handles lexicalization. This is done based on an analysis of the corpus. We manually counted the words and phrases that are used for each attribute, and selected those that were used most often. Only the training set was used for the word counts. The results can be found in Tables 3 and 4. The idea behind this is that extracting the lexicalization from the corpus makes the generated descriptions more human-like, as they use the words actually chosen by humans.

For the furniture domain, almost all words in the corpus were the same as the direct translations. An exception was the orientation attribute, which was difficult to find the right phrases for. We counted 40 different ways of expressing the orientation in 66 mentions of this attribute. This matches the findings of Hervás et al. (2013), who found 79 different lexicalizations for 127 mentions of the orientation attribute in the English TUNA corpus. Many people expressed the left/right orientation property as "from the side". Because this expression removes some information, we decided not to use it. Some expressions of orientation used properties of the object. Examples are "met de leuning links" (with the handrail left). Since this information is not available in our generation data, these cannot be used. We used the phrases shown in Table 3, because they do not have these problems, and are similar to many variations that were seen in the corpus.

This algorithm also improved the baseline algorithm in some other respects. In the baseline algorithm, the combination of hasHair.1 and hairColour was not handled correctly, resulting in phrases such as "de donkerharige man met haar" (the dark-haired man with hair). This problem was fixed by combining the hairColour and hasHair attributes. For example, the combination hasHair=1 and hairColour=dark was realized as "de man met donker haar" (the man with dark hair) instead of "de donkerharige man met haar". When a hair colour is present without the hasHair attribute, it is ignored.

Algorithm 3: Generated templates

The final algorithm uses a method of automatically extracting the order of attributes from the corpus. Again, we only used the training set. The program creates 'templates' by analysing each description in the training set. These templates are not the same as in Brugman et al. (2009). Their templates were manually generated and contained the entire description, including function words such as "with" or "and". Our templates only contain information about the order; the description itself is realized in a way similar to the previous algorithm. In combination with the fact that our templates are extracted automatically, this makes our approach more generic than that of Brugman et al.

For each combination of attributes, the program counts which order was used. For example, for the combination of hasTie, hasShirt, type and hasSuit, most people used the order *type - hasShirt - hasTie - hasSuit*, which results in the template {type}{hasShirt}{hasTie}{hasSuit}. This then results in the description "de man met een witte blouse, een stropdas en een pak" (the man with a shirt, a tie and a suit).

The most used order is stored and will be chosen when the algorithm has to realize a description using that combination of attributes. Like in the 'improved lex' algorithm, the hairColour attribute is assumed to belong to the hasHair attribute, and is ignored when it occurs on its own.

Compared to the 'improved lex' algorithm, the templates algorithm allows for more flexible attribute ordering. For example, it allows the hasHair attribute to be mentioned both before and after the word "man". As a result, both "de donkerharige man" (de dark-haired man) and"de man met donker haar" (the man with dark hair) are possible. The same holds for "de kale man" (the bald man) and "de man met een kaal hoofd" (the man with a bald head). Both orders are present in the templates.

If no template is found for a certain combination, the previous algorithm is used as a fallback.

Evaluation method

Automatic evaluation

For the evaluation, we used the program `teval` (Gatt et al., 2009) to evaluate the accuracy, string edit distance and BLEU-3 score. Each algorithm was evaluated on the test set, with two types of input: once using the original attributes from the D-TUNA corpus, and once using generated attributes from the GRAPH-algorithm (Krahmer et al., 2003). These were evaluated separately to see the influence of attribute selection on the scores. Using the original attributes, we can realize descriptions that use the same attributes as to the original, and we will not get lower scores for entirely different descriptions where the algorithm chose different attributes. On the other hand, the realizer should also work well with the automatic attribute selection. Therefore we evaluate with both the original attributes, and the attributes generated by GRAPH.

Human evaluation

For the human evaluation, we realized the test set using the three algorithms. To see whether the attribute selection step had any influence on the scores, both the original and generated attributes were used. The original human expressions were also included in the evaluation, giving 2x7x40 = 560 phrases. Because of similarities between the algorithms, the same description would often appear more than once. These double descriptions were not removed. Each of the test subjects had to score five descriptions from each realizer on clarity and fluency. The scores were between 1 (bad) and 5 (great). The test form had a list of 70 descriptions, with underneath each description room to score that description for clarity and fluency.

Eight people participated in the evaluation experiment, each scoring 70 descriptions. Three test subjects did the evaluation on paper, while five used an online spreadsheet which looked similar to the offline version. The test subjects were between 16 and 53 years old. All were native speakers of Dutch.

The form started with 5 phrases of one algorithm, followed by 5 phrases of the next algorithm. Each set of 70 descriptions started with a different algorithm. This was done to prevent the placing of each algorithm from influencing the scores.

Results

The results of both the automatic and human evaluation are also shown in Tables 5 and 6.

Automatic evaluation

The results show that using the original attributes as input results in higher similarity scores than when using generated attributes. They also show that almost every time, the new version of the algorithm is at least as good as the previous one.

For the furniture set, the 'improved lex' and the template algorithm show exactly the same scores. We also see that the scores for the generated attributes show as much improvement as the scores for the original attributes, even though they are much lower. For the people set, this is not the case. The differences between the baseline and template algorithms are minimal for generated attributes. The scores for the 'improved lex' and template algorithms are even exactly the same, even though they are completely different when we use the original attributes.

Human evaluation

In the human evaluation, we see a slight increase of the fluency scores in the furniture domain. There is no large difference between the scores of generated and original attributes. A noticeable result is that the template algorithm received slightly lower scores than the 'improved lex' algorithm.

In the people domain, the clarity scores remain more or less the same between different algorithms. Both the fluency and clarity scores are higher for the generated attributes than for the original attributes in the baseline, but this difference

	Baseline		Improved		Templates		Real
	generated	original	generated	original	generated	original	
String accuracy	0.1	0.15	0.1	0.15	0.1	0.15	-
Mean edit distance	6.60	6.30	6.40	6.00	6.40	6.00	-
BLEU-3 score	0.2202	0.2504	0.3138	0.3662	0.3138	0.3662	-
Clarity mean	4.33	4.30	4.28	4.35	4.15	4.23	3.9
Clarity≥4	80%	85%	83%	83%	78%	80%	68%
Fluency mean	3.98	4.33	4.05	4.15	4.08	3.93	3.83
Fluency≥4	75%	90%	73%	85%	78%	68%	79%

Table 5: The results of the evaluation in the furniture domain

	Baseline		Improved		Templates		Real
	generated	original	generated	original	generated	original	
String accuracy	0	0	0	0	0	0	-
Mean edit distance	6.45	5.925	6.325	5.7	6.325	5.2	-
BLEU-3 score	0.1839	0.2847	0.1868	0.3319	0.1868	0.3745	-
Clarity mean	4.65	4.30	4.40	4.60	4.55	4.48	4.18
Clarity≥4	95%	83%	93%	95%	95%	90%	78%
Fluency mean	4.70	3.98	4.70	4.63	4.80	4.63	3.70
Fluency≥4	95%	65%	95%	93%	100%	90%	65%

Table 6: The results of the evaluation in the people domain

seems to disappear for the last two algorithms. The 'improved lex' and template algorithms perform very well, with the template algorithm even getting no scores lower than 4 for fluency with generated attributes.

Finally, we notice that the scores for the human-written descriptions are lower than the scores for the computer-generated descriptions.

Fallback

To evaluate the coverage of the template algorithm, we checked how many times it used the 'improved lex' algorithm as fallback when no template was found. This happened 15 times for the people domain when using generated attributes. When using the original attributes, it only happened twice. This difference can be explained by the fact that the GRAPH-algorithm often chooses a combination of attributes that a human would not choose.

For the furniture domain, the output of the template algorithm always matched the output of the 'improved lex' algorithm.

Discussion

Automatic evaluation

We notice that the 'improved lex' and template algorithms both create exactly the same descriptions for the furniture set. This makes sense, as the adjective order for furniture descriptions appears to be almost fixed. In fact, only 13 of the 140 descriptions from the corpus deviated from the *size-colour-type-orientation* order. None of these resulted in a generated template, as they were far outnumbered by the 'normal' order. The higher scores for the original attributes can also be easily explained: similarity in the attributes results in more similar phrases. The same applies to the people domain.

The poor improvement when using generated attributes for the people domain is harder to explain, especially since the original attribute method has sharply rising scores. Perhaps this is due to the fact that the attribute selection step usually selects very minimal attributes, while people usually have a lot more to say. Descriptions with only one attribute often result in the same or very similar descriptions across different algorithms.

Human evaluation

The improved algorithms resulted in higher fluency scores most of the time. One exception was the template realizer in the furniture domain, which scored much lower than the 'improved lex' algorithm. We cannot explain this, as both algorithms should have the exact same output. A possible cause is random variation due to the subjectivity of the human ratings. The clarity scores did not change much. This could be explained by a ceiling effect: almost all generated descriptions were clear enough according to the test subjects.

For the baseline algorithm, the scores for generated attributes were higher than for original attributes. This makes sense: the generated attributes were usually simpler. In a simple algorithm, less data means less that can go wrong. This difference disappeared for the more advanced algorithms.

It seems odd that the human-written descriptions received such low scores. This can be explained by the fact that humans are more likely to produce strange or minimal descriptions, e.g. writing only "bril" (glasses) when describing a man with glasses. The computer-generated descriptions were more consistent.

Conclusions

We described the development process of a Dutch realizer of referring expressions using the data of the D-TUNA corpus. The D-TUNA corpus is a Dutch corpus that consists of descriptions written by human test subjects, who had to uniquely describe a target object among distractor objects. The corpus has two domains: people and pieces of furniture. The goal of the research was to develop an algorithm that can realize a description using data from the corpus to make the description more human-like. The final system consists of three components:

- A simple realization routine that creates a natural language description given a template and word choices.
- A lexicalization method that chooses the words that were most often used in the D-TUNA corpus.
- An attribute ordering method that chooses the ordering most seen in the corpus to create a template.

To better evaluate the effect of each component, we created three algorithms, each being an improvement over the last. The first only used the simple realization routine, using direct translations of the attribute values for lexicalization, and a fixed template for each domain. The second added the words most seen in the corpus. The final algorithm created new templates for each combination of attributes, using an attribute order extracted from the corpus.

We evaluated the three algorithms for string accuracy, mean edit distance and BLEU-3 score. A human evaluation was also performed, in which eight test subjects each rated 70 descriptions for fluency and clarity. We learned that both corpus-based lexicalization and attribute ordering had a positive effect on string similarity and human-rated fluency for the people domain. For the furniture domain, corpus-based attribute ordering had no effect as the ordering extracted from the corpus exactly matched the theoretical ordering of attributes. This is caused by the simplicity of the furniture domain. We expect that for more complex realistic applications, our method for corpus-based attribute ordering can have a positive effect.

In the future, our method could be improved by developing an automatic system for extracting lexical choice from the corpus, as this was currently done manually. This could decrease the time needed to develop referring expression generation for a new domain.

References

Bohnet, B. (2008). The fingerprint of human referring expressions and their surface realization with graph transducers (IS-FP, IS-GT, IS-FP-GT). In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG 2008, pages 207–210.

Brugman, I., Theune, M., Krahmer, E., and Viethen, J. (2009). Realizing the costs: Template-based surface realisation in the GRAPH approach to referring expression generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG 2009, pages 183–184.

Di Fabbrizio, G., Stent, A. J., and Bangalore, S. (2008). Referring expression generation using speaker-based attribute selection and trainable realization (ATTR). In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG 2008, pages 211–214.

Gatt, A., Belz, A., and Kow, E. (2009). The TUNA-REG challenge 2009: overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG 2009, pages 174–182.

Geerts, G., Haeseryn, W., de Rooij, J., and van der Toorn, M. (1984). *Algemene Nederlandse Spraakkunst*. Wolters-Noordhoff, Groningen and Wolters, Leuven.

Gervás, P., Hervás, R., and León, C. (2008). NIL-UCM: Most-frequent-value-first attribute selection and best-scoring-choice realization. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG 2008, pages 215–218.

Hervás, R., Francisco, V., and Gervás, P. (2013). Assessing the influence of personal preferences on the choice of vocabulary for natural language generation. *Information Processing & Management*, 49(4):817 – 832.

Koolen, R. and Krahmer, E. (2010). The D-TUNA corpus: A Dutch dataset for the evaluation of referring expression generation algorithms. In *International Conference on Language Resources and Evaluation*, LREC 2010, pages 122–127.

Krahmer, E., van Erk, S., and Verleg, A. (2003). Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Pereira, D. B. and Paraboni, I. (2008). From TUNA attribute sets to Portuguese text: a first report. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG 2008, pages 232–233.

van Deemter, K., van der Sluis, I., and Gatt, A. (2006). Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 130–132.