

# INLG 2010

Proceedings of the Sixth  
International Natural Language  
Generation Conference

July 7 - 9, 2010

Trim, Co. Meath, Ireland



Programme Chairs: John Kelleher, Brian Mac Namee & Ielka van der Sluis  
Generation Challenge Chairs: Anja Belz, Albert Gatt & Alexander Koller

**Cover image:** Trim Castle by photographer Andrew Parnell used under the Creative Commons Attribution 2.0 Generic License.  
<http://flickr.com/photos/andrewparnell/>  
<http://flickr.com/photos/andrewparnell/365752330/>

# **INLG 2010**

## **Proceedings of the Sixth International Natural Language Generation Conference**

Program Chairs:

John Kelleher, Brian Mac Namee & Ielka van der Sluis

Generation Challenge Chairs:

Anja Belz, Albert Gatt & Alexander Koller

July 7–9, 2010

Trim, Co. Meath, Ireland

Hosted by the University of Dublin, Trinity College and the Dublin Institute of Technology  
Endorsed by the ACL Special Interest Group on Natural Language Generation (SIGGEN)  
Sponsored by the Centre for Next Generation Localisation and Science Foundation Ireland



© 2010 The Association for Computational Linguistics

## Table of Contents

Preface .....	v
Conference Organisation .....	vii
Conference Program .....	ix

### Keynote Speakers

Adapting Generation to Addressees: What Drives Audience Design? .....	2
<i>Susan E. Brennan</i>	
Ontologies and Text: Can NLG Bridge the Gap? .....	3
<i>Richard Power</i>	

### Full Papers

Comparing Rating Scales and Preference Judgements in Language Evaluation .....	7
<i>Anja Belz &amp; Eric Kow</i>	
A Discourse-Aware Graph-Based Content-Selection Framework .....	17
<i>Seniz Demir, Sandra Carberry &amp; Kathleen F. McCoy</i>	
Generating Referring Expressions with Reference Domain Theory .....	27
<i>Alexandre Denis</i>	
Hierarchical Reinforcement Learning for Adaptive Text Generation .....	37
<i>Nina Dethlefs &amp; Heriberto Cuayahuitl</i>	
Tense and Aspect Assignment in Narrative Discourse .....	47
<i>David Elson &amp; Kathleen McKeown</i>	
Textual Properties and Task-based Evaluation: Investigating the Role of Surface Properties, Structure and Content .....	57
<i>Albert Gatt &amp; Francois Portet</i>	
Situated Reference in a Hybrid Human-Robot Interaction System .....	67
<i>Manuel Giuliani, Mary Ellen Foster, Amy Isard, Colin Matheson, Jon Oberlander &amp; Alois Knoll</i>	
Towards a Programmable Instrumented Generator .....	77
<i>Chris Mellish</i>	

Using Semantic Web Technology to Support NLG. Case Study: OWL finds RAGS . . . . .	85
<i>Chris Mellish</i>	
Natural Reference to Objects in a Visual Domain . . . . .	95
<i>Margaret Mitchell, Kees van Deemter &amp; Ehud Reiter</i>	
Generating and Validating Abstracts of Meeting Conversations: a User Study . . . . .	105
<i>Gabriel Murray, Giuseppe Carenini &amp; Raymond Ng</i>	
Charting the Potential of Description Logic for the Generation of Referring Expressions . . . . .	115
<i>Yuan Ren, Kees van Deemter &amp; Jeff Z. Pan</i>	
Complex Lexico-syntactic Reformulation of Sentences Using Typed Dependency Representations . . . . .	125
<i>Advaith Siddharthan</i>	
Towards an Extrinsic Evaluation of Referring Expressions in Situated Dialogs . . . . .	135
<i>Philipp Spanger, Ryu Iida, Takenobu Tokunaga, Asuka Terai &amp; Naoko Kuriyama</i>	
Harvesting Re-usable High-level Rules for Expository Dialogue Generation	145
<i>Svetlana Stoyanchev &amp; Paul Piwek</i>	
Feature Selection for Fluency Ranking . . . . .	155
<i>Daniël de Kok</i>	

## Short Papers

Extracting Parallel Fragments from Comparable Corpora for Data-to-text Generation . . . . .	167
<i>Anja Belz &amp; Eric Kow</i>	
Generating Natural Language Descriptions of Z Test Cases . . . . .	173
<i>Maximiliano Cristià &amp; Brian Plüss</i>	
Applying Semantic Frame Theory to Automate Natural Language Template Generation From Ontology Statements . . . . .	179
<i>Dana Dannélls</i>	
'If you've heard it, you can say it' - Towards an Account of Expressibility	185
<i>David McDonald &amp; Charlie Greenbacker</i>	
Cross-linguistic Attribute Selection for REG: Comparing Dutch and English . . . . .	191
<i>Mariet Theune, Ruud Koolen &amp; Emiel Krahmer</i>	

Grouping Axioms for More Coherent Ontology Descriptions . . . . .	197
<i>Sandra Williams &amp; Richard Power</i>	
Paraphrase Generation as Monolingual Translation: Data and Evaluation	203
<i>Sander Wubben, Antal van den Bosch &amp; Emiel Krahmer</i>	
Anchor-Progression in Spatially Situated Discourse: a Production Experiment . . . . .	209
<i>Hendrik Zender, Christopher Koppermann, Fai Greeve &amp; Geert-Jan Kruijff</i>	

## INLG Generation Challenges 2010

Preface . . . . .	217
<i>Anja Belz, Albert Gatt &amp; Alexander Koller</i>	
The GREC Challenges 2010: Overview and Evaluation Results . . . . .	219
<i>Anja Belz &amp; Eric Kow</i>	
Named Entity Generation Using Sampling-based Structured Prediction . .	230
<i>Guillaume Bouchard</i>	
Poly-co: An Unsupervised Co-reference Detection System . . . . .	233
<i>Éric Charton, Michel Gagnon &amp; Benoit Ozell</i>	
JU_CSE_GREC10: Named Entity Generation at GREC 2010 . . . . .	235
<i>Amitava Das, Tanik Saikh, Tapabrata Mondal &amp; Sivaji Bandyopadhyay</i>	
The UMUS System for Named Entity Generation at GREC 2010 . . . . .	237
<i>Benoit Favre &amp; Bernd Bohnet</i>	
UDeI: Refining a Method of Named Entity Generation . . . . .	239
<i>Charles Greenbacker, Nicole Sparks, Kathleen McCoy &amp; Che-Yu Kuo</i>	
UDeI: Named Entity Recognition and Reference Regeneration from Surface Text . . . . .	241
<i>Nicole Sparks, Charles Greenbacker, Kathleen McCoy &amp; Che-Yu Kuo</i>	
Report on the Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2) . . . . .	243
<i>Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johann Moore &amp; Jon Oberlander</i>	
The First Question Generation Shared Task Evaluation Challenge . . . . .	251
<i>Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoy- anchev &amp; Christian Moldovan</i>	
Generation Under Uncertainty . . . . .	255
<i>Oliver Lemon, Sridhi Janarthanam &amp; Verena Rieser</i>	

Helping Our Own: Text Massaging for Computational Linguistics as a New Shared Task . . . . .	261
<i>Robert Dale &amp; Adam Kilgarriff</i>	
Finding Common Ground: Towards a Surface Realisation Shared Task . . .	267
<i>Anja Belz, Mike White, Josef van Genabith, Deirdre Hogan &amp; Amanda Stent</i>	

## **Appendices**

Author Index . . . . .	275
------------------------	-----



## Preface

It gives us great pleasure to introduce the technical program of the Sixth International Natural Language Generation Conference (INLG 2010), the biennial meeting of the ACL Special Interest Group in Natural Language Generation (SIGGEN). The INLG conference provides the premier forum for the discussion, dissemination and archiving of research and results in the field of natural language generation. Previous INLG conferences have been held in the USA, Australia, the UK and Israel. Prior to 2000, INLG meetings were held as international workshops with a history stretching back to 1983. In 2010, on behalf of SIGGEN, INLG is being co-hosted by Trinity College Dublin and the Dublin Institute of Technology; and held in Trim Castle Hotel, Trim, Co. Meath, Ireland.

The INLG 2010 programme consists of presentations of substantial, original, and previously unpublished results on all topics related to natural language generation. This year we received 50 submissions (36 full papers and 14 short papers) from 18 different countries from around the world. As in previous years, each submission was reviewed by at least three members of an international programme committee of leading researchers in the field. Based on these reviews 16 submissions were accepted as full papers and 8 as short papers (4 papers were withdrawn). The accepted papers are of the highest quality and cover all of the major aspects of natural language generation.

This year, the conference programme includes two keynote speakers. Susan E. Brennan, Professor of Psychology at Stony Brook University, will speak on “Adapting Generation to Addressees: What Drives Audience Design?” and Richard Power of The Open University will present a talk entitled “Ontologies and Text: Can NLG Bridge the Gap?”. This year we are also delighted to host the 2010 Generation Challenges organised by Anja Belz, Albert Gatt and Alexander Koller. This is a part of INLG that has been growing in importance over the last number of conferences and is a great addition to the event.

The organising committee would like to offer their thanks to our invited speakers for agreeing to join us, the organisers of INLG 2008 for their enormous help, the SIGGEN board for allowing us host the conference and for their assistance, Priscilla Rasmussen at ACL and Alena Moison at TCD for handling finances, the programme committee for their dedicated work, and, most of all, the authors of all submitted papers. We have also received generous sponsorship from the Centre for Next Generation Localisation and Science Foundation Ireland for which we are extremely grateful.

Finally, we would like to welcome you to Trim and hope that you have an enjoyable and inspiring visit. We will leave you with an Irish proverb in the spirit of INLG: *Tír gan teanga, tír gan anam.*

The INLG 2010 Organising Committee  
John Kelleher, Brian Mac Namee & Ielka van der Sluis



## Conference Organization

### Programme Chairs

John Kelleher, Dublin Institute of Technology, Ireland  
Brian Mac Namee, Dublin Institute of Technology, Ireland  
Ielka van der Sluis, Trinity College Dublin, Ireland

### Generation Challenge Chairs

Anja Belz, University of Brighton, UK  
Albert Gatt, University of Malta, Malta  
Alexander Koller, Universitaet des Saarlandes, Germany

### Programme Committee

John Bateman, University of Bremen, Germany  
Anja Belz, University of Brighton, UK  
Bernd Bohnet, University Stuttgart, Germany  
Stephan Busemann, DFKI GmbH, Germany  
Christian Chiarcos, Universitaet Potsdam, Germany  
Norman Creaney, University of Ulster, UK  
Robert Dale, Macquarie University, Australia  
Kees van Deemter, University of Aberdeen, UK  
David DeVault, USC Institute for Creative Technologies, US  
Barbara Di Eugenio, University of Illinois, US  
Roger Evans, University of Brighton, UK  
Jennifer Foster, Dublin City University, Ireland  
Mary Ellen Foster, Heriot Watt University, Edinburgh, UK  
Claire Gardent, CNRS/LORIA, France  
Albert Gatt, University of Malta, Malta  
Josef van Genabith, Dublin City University, Ireland  
Pablo Gervas, Universidad Complutense de Madrid, Spain  
Markus Guhe, University of Edinburgh, UK  
Svetlana Hensman, Dublin Institute of Technology, Ireland  
Alexander Koller, Universitaet des Saarlandes, Germany  
Alistair Knott, University of Otago, New Zealand  
Emiel Krahmer, Tilburg University, The Netherlands  
Ivana Kruijff-Korbayova, Saarland University, Germany  
Oliver Lemon, Heriot Watt University, Edinburgh, UK  
James Lester, North Carolina State University, US  
Keith Vander Linden, Calvin College, US

Kathleen McCoy, University of Delaware, US  
David McDonald, BBN Technologies, US  
Chris Mellish, University of Aberdeen, UK  
Johanna Moore, University of Edinburgh, UK  
Cecile Paris, CSIRO ICT Centre, Australia  
Paul Piwek, the Open University, UK  
Ehud Reiter, University of Aberdeen, UK  
Graeme Ritchie, University of Aberdeen, UK  
Advaith Siddharthan, University of Aberdeen, UK  
Yaji Sripada, University of Aberdeen, UK  
Matthew Stone, Rutgers, US  
Manfred Stede, Universitaet Potsdam, Germany  
Amanda Stent, AT&T Labs, US  
Kristina Striegnitz, Union College, US  
Michael Strube, EML Research, Germany  
Takenobu Tokunaga, Tokyo Institute of Technology, Japan  
Mariet Theune, University of Twente, The Netherlands  
Sebastian Varges, DISI Trento, Italy  
Carl Vogel, Trinity College Dublin, Ireland  
Michael White, Ohio State University, US  
Sandra Williams, the Open University, UK  
Tiejun Zhao, Harbin Institute of Technology, China

## **Subreviewers**

Virginia Francisco  
Raquel Hervás  
Carlos Leàn  
Mo Yu  
Conghui Zhu

## Wednesday, July 7<sup>th</sup>, 2010

12:00 - 13:30 Buffet lunch and Check-in

### Session 1: Invited Talk

13:30 - 13:45 Opening Remarks

13:45 - 14:45 Ontologies and text: Can NLG bridge the gap? (*Invited Talk*)  
*Richard Power*

14:45 - 15:00 **Break**

### Session 2: Ontology Based Generation

15:00 - 15:30 Using Semantic Web Technology to Support NLG.  
Case study: OWL finds RAGS.  
*Chris Mellish*

15:30 - 16:00 Charting the Potential of Description Logic for the  
Generation of Referring Expressions.  
*Yuan Ren, Kees van Deemter and Jeff Z. Pan*

16:00 - 16:30 Generating and Validating Abstracts of Meeting  
Conversations: a User Study.  
*Gabriel Murray, Giuseppe Carenini and Raymond Ng*

16:30 - 16:45 **Break**

### Session 3: Sentence Level Generation and Machine Learning in NLG

16:45 - 17:15 Complex Lexico-syntactic Reformulation of Sentences  
Using Typed Dependency Representations.  
*Advaith Siddharthan*

17:15 - 17:45 Feature Selection for Fluency Ranking.  
*Daniel de Kok*

17:45 - 18:15 Hierarchical Reinforcement Learning for Adaptive Text Generation.  
*Nina Dethlefs and Heriberto Cuayahuitl*

Evening Banquet and drinks and music at the Hotel Bar

## Thursday, July 8<sup>th</sup>, 2010 (*Morning*)

### Generation Challenges

08:30 - 09:00 Preparation for Generation Challenge Poster Session

#### GC Session 1: Shared Task Reports

*(chaired by Albert Gatt)*

09:00 - 09:10 Introduction (*Albert Gatt*)

09:10 - 09:40 GREC'10 results presentation

*Anja Belz*

09:40 - 10:05 GIVE-2 results presentation

*Alexander Koller*

10:05 - 10:20 Question Generation presentation

*Vasile Rus*

10:20 - 10:50 **GC Poster Session and Tea/Coffee Break**

#### GC Session 2: Invited Talk

10:50 - 11:35 What speakers do and don't do to communicate successfully.

*Victor Ferreira*

#### GC Session 3: New Shared Task Proposals

*(chaired by Anja Belz)*

11:35 - 11:55 Generation Under Uncertainty

*Oliver Lemon*

11:55 - 12:15 Text Improvement

*Robert Dale*

12:15 - 12:35 Surface Realisation

*Mike White*

12: 35 - 13:30 **Lunch**

#### Generation Challenge Working Lunch

Table 1: Generation Under Uncertainty; chair: Oliver Lemon

Table 2: Text Improvement; chair: Robert Dale

Table 3: Surface Realisation; chair: Mike White

## Thursday, July 8<sup>th</sup>, 2010 (*Afternoon/Evening*)

### Session 4: Evaluation in NLG and Poster Introductions

- 13:30 - 14:00 Towards a Programmable Instrumented Generator.  
*Chris Mellish*
- 14:00 - 14:30 Comparing Rating Scales and Preference Judgements  
in Language Evaluation.  
*Anja Belz and Eric Kow*
- 14:30 - 15:00 Textual properties and task-based evaluation: Investigating  
the role of surface properties, structure and content.  
*Albert Gatt and Francois Portet*
- 15:00 - 15:15 Poster Introductions
- 15:15 - 17:15 **Activity break**
- 18:00 - late INLG Poster session with drinks and dinner and music!

## Friday, July 9<sup>th</sup>, 2010

### Session 5: Invited Talk

09:00 - 10:00 Adapting Generation to Addressees: What Drives Audience Design?  
*Susan E. Brennan*

10:00 - 10:15 **Break**

### Session 6: Situated Reference

10:15 - 10:45 Situated Reference in a Hybrid Human-Robot Interaction System.  
*Manuel Giuliani, Mary Ellen Foster, Amy Isard,  
Colin Matheson, Jon Oberlander and Alois Knoll*

10:45 - 11:15 Natural Reference to Objects in a Visual Domain.  
*Margaret Mitchell, Kees van Deemter and Ehud Reiter*

11:15 - 11:45 Generating Referring Expressions with Reference Domain Theory.  
*Alexandre Denis*

11:45 - 12:15 Towards an extrinsic evaluation of referring expressions  
in situated dialogs.  
*Philipp Spanger, Ryu Iida, Takenobu Tokunaga,  
Asuka Terai and Naoko Kuriyama*

12:15 - 13:30 **Lunch**

### Session 7: Discourse/Dialogue Generation

13:30 - 14:00 Harvesting Re-usable High-level Rules for  
Expository Dialogue Generation.  
*Paul Piwek and Svetlana Stoyanchev*

14:00 - 14:30 A Discourse-Aware Graph-Based Content-Selection Framework.  
*Seniz Demir, Sandra Carberry and Kathleen F. McCoy*

14:30 - 15:00 Tense and Aspect Assignment in Narrative Discourse.  
*David Elson and Kathleen McKeown*

15:00 - 15:15 **Closing Remarks**

16:00 **Bus departs for Dublin**



# Keynote Speakers

## Adapting Generation to Addressees: What Drives Audience Design?

Susan E. Brennan

Stony Brook University  
New York, USA  
[susan.brennan@sunysb.edu](mailto:susan.brennan@sunysb.edu)

**Abstract:** Utterances are enormously variable in the forms they take. Although variability is often treated as noise to be normalized or filtered out, some who study spoken dialog probably suspect that this variability is meaningful. In this talk I will present experimental data about partner-specific variability, or audience design. No one disputes that audience design exists, but there is debate about how and why it emerges and whether it matters. After discussing some systematic ways in which speakers adapt their utterances to addressees, I will consider: What drives this adaptation? How does it affect processing by addressees? And what are the implications for natural language generation?

**Bio:** Susan Brennan is Professor of Psychology at Stony Brook University and is also affiliated with the Departments of Linguistics and Computer Science. She received a Ph.D. in Cognitive Psychology from Stanford University with a focus on psycholinguistics; an M.S. from the MIT Media Lab, where she worked on computer-generated caricatures and mediated communication; and a B.A. in cultural anthropology from Cornell University. She uses eyetracking and other behavioral techniques to study language processing by interacting partners.

## Ontologies and Text: Can NLG Bridge the Gap?

Richard Power

The Open University  
Milton Keynes, United Kingdom  
`r.power@open.ac.uk`

**Abstract:** Ontologies are akin to technical documents in that they describe domain knowledge, but they express this content very differently, in formal languages like OWL designed for use by machines, not people. During the last decade, interest has grown in the task of mapping from OWL to controlled fragments of natural language, thus providing a niche for NLG in which we are at last agreed on the formal specification of the input.

The aim of the talk is to compare ontologies and their textual counterparts (e.g., technical dictionaries, encyclopedias) at several linguistic levels. After looking at their usage (pragmatic level), we will consider terminology (roughly, word level), statements (sentence level), and groups of related statements (discourse level). The question is whether we can find similar levels in the organisation of OWL ontologies, thus allowing a mapping from ontologies to texts that can be exploited by NLG systems.

**Bio:** Richard Power has a B.A. in Psychology from the University of Sheffield, and a PhD from the University of Edinburgh for research on generating conversation. From 1975-78 he worked as a postdoc at the University of Sussex, on topics including automatic learning of numeral systems. He then moved to Padua, Italy, where he taught English in the Psychology department for some years, and worked as chief scientist and knowledge engineer for a Milan-based Artificial Intelligence company. In 1993 he returned to the UK and joined the Information Technology Research Centre at the University of Brighton. Since 2005 he has been senior lecturer in the Department of Computing at the Open University.

His research interests since 1993 have focussed on two areas in NLG: applications of Constraint Logic Programming (especially in the ICONOCLAST project), and natural language tools for knowledge editing (the WYSIWYM systems). He is currently working on the development of NLG-based tools for viewing and editing knowledge on the semantic web (SWAT project).



# Full Papers



# Comparing Rating Scales and Preference Judgements in Language Evaluation

Anja Belz      Eric Kow

Natural Language Technology Group  
School of Computing, Mathematical and Information Sciences  
University of Brighton  
Brighton BN2 4GJ, UK  
{asb, eykk10}@bton.ac.uk

## Abstract

Rating-scale evaluations are common in NLP, but are problematic for a range of reasons, e.g. they can be unintuitive for evaluators, inter-evaluator agreement and self-consistency tend to be low, and the parametric statistics commonly applied to the results are not generally considered appropriate for ordinal data. In this paper, we compare rating scales with an alternative evaluation paradigm, preference-strength judgement experiments (PJE), where evaluators have the simpler task of deciding which of two texts is better in terms of a given quality criterion. We present three pairs of evaluation experiments assessing text fluency and clarity for different data sets, where one of each pair of experiments is a rating-scale experiment, and the other is a PJE. We find the PJE versions of the experiments have better evaluator self-consistency and inter-evaluator agreement, and a larger proportion of variation accounted for by system differences, resulting in a larger number of significant differences being found.

## 1 Introduction

Rating-scale evaluations, where human evaluators assess system outputs by selecting a score on a discrete scale, are the most common form of human-assessed evaluation in NLP. Results are typically presented in rank tables of means for each system accompanied by means-based measures of statistical significance of the differences between system scores.

NLP system evaluation tends to involve sets of systems, rather than single ones (evaluations tend to at least incorporate a baseline or, more rarely, a topline system). The aim of system evaluation is

to gain some insight into which systems are better than which others, in other words, the aim is inherently relative. Yet NLP system evaluation experiments have generally preferred rating scale experiments where evaluators assess each system's quality in isolation, in absolute terms.

Such rating scales are not very intuitive to use; deciding whether a text deserves a 5, a 4 or a 3 etc. can be difficult. Furthermore, evaluators may ascribe different meanings to scores and the distances between them. Individual evaluators have different tendencies in using rating scales, e.g. what is known as 'end-aversion' tendency where certain individuals tend to stay away from the extreme ends of scales; other examples are positive skew and acquiescence bias, where individuals make disproportionately many positive or agreeing judgements; see e.g. Choi and Pak, (2005).

It is not surprising then that stable averages of quality judgements, let alone high levels of agreement, are hard to achieve, as has been observed for MT (Turian et al., 2003; Lin and Och, 2004), text summarisation (Trang Dang, 2006), and language generation (Belz and Reiter, 2006). It has even been demonstrated that increasing the number of evaluators and/or data can have no stabilising effect at all on means (DUC literature).

The result of a rating scale experiment is ordinal data (sets of scores selected from the discrete rating scale). The means-based ranks and statistical significance tests that are commonly presented with the results of RSEs are not generally considered appropriate for ordinal data in the statistics literature (Siegel, 1957). At the least, "a test on the means imposes the requirement that the measures must be additive, i.e. numerical" (Siegel, 1957, p. 14). Parametric statistics are more powerful than non-parametric alternatives, because they make a number of strong assumptions (including that the data is numerical). If the assumptions are violated then the risks is that the significance of results is

overestimated.

In this paper we explore an alternative evaluation paradigm, Preference-strength Judgement Experiments (PJE). Binary preference judgements have been used in NLP system evaluation (Reiter et al., 2005), but to our knowledge this is the first systematic investigation of preference-strength judgements where evaluators express, in addition to their preference (*which system do you prefer?*), also the strength of their preference (*how strongly do you prefer the system you prefer?*). It seems intuitively convincing that it should be easier to decide which of two texts is clearer than to decide whether a text’s clarity deserves a 1, 2, 3, 4 or 5. However, it is less clear whether evaluators are also able to express the strength of their preference in a consistent fashion, resulting not only in good self-consistency, but also in good agreement with other evaluators.

We present three pairs of directly comparable RSE and PJE evaluations, and investigate how they compare in terms of (i) the amount of variation accounted for by differences between systems (the more the better), relative to the amount of variation accounted for by other factors such as evaluator and arbitrary text properties (the less the better); (ii) inter-evaluator agreement, (iii) evaluator self-consistency, (iv) the number of significant differences identified, and (v) experimental cost.

## 2 Overview of Experiments

In the following three sections we present the design and results of three pairs of evaluations. Each pair consists of a rating-scale experiment (RSE) and a preference-strength judgement experiment (PJE) that differ only in the rating method they employ (relative ratings in the PJE and absolute ratings in the RSE).<sup>1</sup> In other words, they involve the same set of system outputs, the same instructions and method of presenting system outputs. Each pair is for a different data domain and system task, the first for generating chains of references to people in Wikipedia articles (Section 3); the second for weather forecast text generation (Section 4); and the third for generating descriptions of images of furniture and faces (Section 5).

All experiments use a Repeated Latin Squares

<sup>1</sup>We are currently preparing an open-source release of the RSE/PJE toolkit we have developed for implementing the experiments described in this paper which automatically generates an experiment, including webpages, given some user-specified parameters and the data to be evaluated.

**Fluency**

- 5. Very good (all parts read well)
- 4. Good (most parts read well)
- 3. Neither good nor poor
- 2. Poor (most parts don't read well)
- 1. Very poor (all parts don't read well)

**Clarity**

- 5. Very good (all parts are clear)
- 4. Good (most parts are clear)
- 3. Neither good nor poor
- 2. Poor (most parts are unclear)
- 1. Very poor (all parts are unclear)

Submit rating

Figure 1: Standardised 1–5 rating scale representation for Fluency and Clarity criteria.

design which ensures that each subject sees the same number of outputs from each system and for each test set item. Following detailed instructions, subjects first do 2 or 3 practice examples, followed by the texts to be evaluated, in an order randomised for each subject. Subjects carry out the evaluation over the internet, at a time and place of their choosing. They are allowed to interrupt and resume (but are discouraged from doing so).

There are subtle differences between the three experiment pairs, and for ease of comparison we provide an overview of the six experiments we investigate in this paper in Table 1. Each of the aspects of experimental design and execution shown in this table is explained and described in more detail in the relevant subsection below, but some of the important differences are highlighted here.

In GREC-NEG PJE, each system is compared with only one other comparisor system (a human-authored topline), whereas in the other two PJE experiments, each system is compared with all other systems for each test data set item.

In the two versions of the METEO evaluation, evaluators were not drawn from the same cohort of people, whereas in the other two evaluation pairs they were drawn from the same cohort. GREC-NEG RSE and METEO RSE used radio buttons (as shown in Figure 1) as the rating-scale evaluation mechanism whereas in TUNA RSE it was an unmarked slider bar. While slightly different names were used for the evaluation criteria in two of the evaluation pairs, Fluency/Readability were explained in very similar terms (*does it read well?*), and Adequacy in TUNA was explained in terms of clarity of reference (*is it clear which entity the description refers to?*), so there are in fact just two evaluation criteria (albeit with different names).

Where we use preference-strength judgements,



Data set	GREC-NEG		METEO		TUNA	
	RSE	PJE	RSE	PJE	RSE	PJE
Criteria names	Fluency, Clarity		Readability, Clarity		Fluency, Adequacy	
Evaluator type	linguistics students		uni staff	ling stud	linguistics students	
Num evaluators	10	10	22	22	8	28
Comparator(s)	–	human topline	–	all systems	–	all systems
Test set size	30		22		112	
N trials	300	300	484	1210	896	3136
Rating tool	radio buttons	slider	radio buttons	slider	slider bar	slider
Range	1–5	–10.0.. + 10.0	1–7	–50.0.. + 50.0	0–100	–50.0.. + 50.0
Numbers visible?	yes	no	yes	no	no	no

Table 1: Overview of experiments with details of design and execution. (Comparator(s) = the other systems against which each system is evaluated.)

the evaluation mechanism is implemented using slider bars as shown at the bottom of Figure 2 which map to a scale  $-X.. + X$ . The evaluator’s task is to express their preference in terms of each quality criterion by moving the pointers on the sliders. Moving the pointer to the left means expressing a preference for the text on the left, moving it to the right means preferring the text on the right; the further to the left/right the slider is moved, the stronger the preference. It was not evident to the evaluators that sliders were associated with numerical values. Slider pointers started out in the middle of the scale (the position corresponding to no preference). If they wanted to leave the pointer in the middle (i.e. if they had no preference for either of the two texts), evaluators had to check a box to confirm their rating (to avoid evaluators accidentally not rating a text and leaving the pointer in the default position).

### 3 GREC-NEG RSE/PJE: Named entity reference chains

#### 3.1 Data and generic design

In our first pair of experiments we used system and human outputs for the GREC-NEG task of selecting referring expressions for people in discourse context. The GREC-NEG data<sup>2</sup> consists of introduction sections from Wikipedia articles about people in which all mentions of people have been annotated by marking up the word strings that function as referential expressions (REs) and annotating them with coreference information as well as syntactic and semantic features. The following is an example of an annotated RE from the corpus:

```
<REF ENTITY="0" MENTION="1" SEMCAT="person" SYNCAT="np"
  SYNFUNC="subj"><REFEX ENTITY="0" REG08-TYPE="name"
```

<sup>2</sup>The GREC-NEG data and documentation is available for download from <http://www.nltg.brighton.ac.uk/home/Anja.Belz>

```
CASE="plain">Sir Alexander Fleming</REFEX> </REF>
(6 August 1881 - 11 March 1955) was a Scottish biologist and pharmacologist.
```

This data was used in the GREC-NEG’09 shared-task competition (Belz et al., 2009), where the task was to create systems which automatically select suitable REs for all references to all person entities in a text.

The evaluation experiments use Clarity and Fluency as quality criteria which were explained in the introduction as follows (the wording of the first is from DUC):

1. **Referential Clarity:** It should be easy to identify who the referring expressions are referring to. If a person is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if a person is referenced, but their identity or relation to the story remains unclear.
2. **Fluency:** A referring expression should ‘read well’, i.e. it should be written in good, clear English, and the use of titles and names should seem natural. Note that the Fluency criterion is independent of the Referential Clarity criterion: a reference can be perfectly clear, yet not be fluent.

The evaluations involved outputs for 30 randomly selected items from the test set from 5 of the 6 systems which participated in GREC-NEG’10, the four baseline systems developed by the organisers, and the original corpus texts (10 systems in total).

#### 3.2 Preference judgement experiment

The human-assessed intrinsic evaluation in GREC’09 was designed as a preference-judgement test where subjects expressed their preference, in terms of the two criteria, for either the original Wikipedia text (human-authored ‘topline’) or the version of it with system-selected referring expressions in it. There were three 10x10 Latin Squares, and a total of 300 trials (with two judgements in each, one for Fluency and one for Clarity) in this evaluation. The subjects were 10

## Ramon Pichot Gironès


Ramon Pichot Gironès (1872 - 1 March 1925) was a Catalan and Spanish artist. He painted in an impressionist style.

He was a good friend of Pablo Picasso and acted as an early mentor to young Salvador Dali. Salvador Dali met Ramon Pichot Gironès in Cadaqués, Spain when Salvador was only 10 years old. Ramon also made many trips to France. Once in a while Salvador and his family would go on a trip with Ramon Pichot and his family.

Ramon Pichot Gironès (1872 - 1 March 1925) was a Catalan and Spanish artist. He painted in an impressionist style.

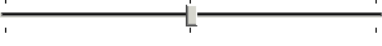
He was a good friend of Pablo Picasso and acted as an early mentor to young Salvador Dali. Salvador Dali met him in Cadaqués, Spain when Salvador was only 10 years old. Ramon also made many trips to France. Once in a while Salvador Dali and his family would go on a trip with Ramon Pichot and his family.

Clarity



move slider or tick here to confirm your rating

Fluency



move slider or tick here to confirm your rating

Figure 2: Example of text pair presented in human intrinsic evaluation of GREC-NEG systems.

native speakers of English recruited from cohorts of students currently completing a linguistics-related degree at Kings College London and University College London.

Figure 2 shows what subjects saw during the evaluation of an individual text pair. The place (left/right) of the original Wikipedia article was randomly determined for each individual evaluation of a text pair. People references are highlighted in yellow/orange, those that are identical in both texts are yellow, those that are different are orange.<sup>3</sup> The sliders are the standardised design described in the preceding section.

### 3.3 Rating scale experiment

Our new experiment used our standardised radio button design for a 1–5 rating scale as shown in Figure 1. We used the same Latin Squares design as for the PJE version, and recruited 10 different evaluators from the same student cohorts at Kings College London and University College London. Evaluators saw just one text in each trial, with the people references highlighted in yellow.

### 3.4 Results and comparative analysis

Measures comparing the results from the two versions of the GREC-NEG evaluation are shown in Table 2. The first row for each experiment type

<sup>3</sup>When viewed in black and white, the orange highlights are the slightly darker ones.

Type	Measure	Clarity	Fluency
RSE (Text (Evaluator	$F_{(9,290)}$	10.975**	35.998**
	N sig diffs	19/45	27/45
	K's W (inter)	.543**	.760**
	avg W (intra)	.5275	.7192
	$F_{(29,270)}$	2.512**	1.825**
	$F_{(9,290)}$	3.998**	.630
PJE (Text (Evaluator	$F_{(9,290)}$	29.539**	26.596**
	N sig diffs	26/45	24/45
	K's W (inter)	.717**	.725**
	avg W (intra)	.6909	.7125
	$F_{(29,270)}$	.910	1.237
	$F_{(9,290)}$	1.237	4.145**

Table 2: GREC-NEG RSE/PJE: Results of analyses looking at effect of System.

shows the F ratio as determined by a one-way ANOVA with the evaluation criterion in question as the dependent variable and System as the factor. F is the ratio of between-groups variability over within-group (or residual) variability, i.e. the larger the value of F, the more of the variability observed in the data is accounted for by the grouping factor, here System, relative to what variability remains within the groups.

The second row shows the number of significant differences out of the possible total, as determined by a Tukey's HSD analysis. Kendall's W (interpretable as a coefficient of concordance) is

a commonly used measure of the agreement between judges and is based on mean rank. It ranges from 0 to 1, and the closer to 1 it is the greater the agreement. The fourth row (K's W, inter) shows the standard W measure, estimating the degree to which the evaluators agreed. The 5th row (K's W, intra) shows the average W for repeated ratings *by the same judge*, i.e. it is a measure of the average self-consistency achieved by the evaluators. Finally, in the last two rows we give F-ratios for Text (test data set item) and Evaluator, estimating the effect these two have independently of System.

The F ratios and numbers of significant differences are very similar in the PJE version, but very dissimilar in the RSE version of this experiment. For Fluency, F is greater in the RSE version than in the PJE version where there appear to be bigger differences between scores assigned by evaluators. However, Kendall's W shows that in terms of mean score ranks, the evaluators agreed to a similar extent in both experiment versions.

Clarity in the RSE version has lower values across the board than the rest of Table 2: it accounts for less of the variation, has fewer significant differences and lower levels of inter-evaluator agreement and self-consistency. If the results from the PJE version were not also available one might be inclined to conclude that there was not as much difference between systems in terms of Clarity as there was in terms of Fluency. However, because Fluency and Clarity have a similarly strong effect in GREC-NEG PJE, it looks instead as though the evaluators found it harder to apply the Clarity criterion in GREC-NEG RSE than Fluency in GREC-NEG RSE, and than Clarity in GREC-NEG PJE.

One way of interpreting this is that it is possible to achieve the same good levels of inter-evaluator and intra-evaluator variation for the Clarity criterion as for Fluency (both as defined and applied within the context of this specific experiment), and that it is therefore worrying that the RSE version does not achieve it.

## 4 METEO RSE/PJE: Weather forecasts

### 4.1 Data

Our second pair of evaluations used the Prodigy-METEO<sup>4</sup> version (Belz, 2009) of the SUMTIME-METEO corpus (Sripada et al., 2002) which contains system outputs and the pairs of wind forecast

<sup>4</sup>The Prodigy-METEO corpus is freely available here: <http://www.nltg.brighton.ac.uk/home/Anja.Belz>

texts and wind data the systems were trained on, e.g.:

```
Data: 1 SSW 16 20 - - 0600 2 SSE - - -
      - NOTIME 3 VAR 04 08 - - 2400
Text: SSW 16-20 GRADUALLY BACKING SSE
      THEN FALLING VARIABLE 4-8 BY
      LATE EVENING
```

The input vector is a sequence of 7-tuples  $\langle i, d, s_{min}, s_{max}, g_{min}, g_{max}, t \rangle$  where  $i$  is the tuple's ID,  $d$  is the wind direction,  $s_{min}$  and  $s_{max}$  are the minimum and maximum wind speeds,  $g_{min}$  and  $g_{max}$  are the minimum and maximum gust speeds, and  $t$  is a time stamp (indicating for what time of the day the data is valid). The wind forecast texts were taken from comprehensive maritime weather forecasts produced by the professional meteorologists employed by a commercial weather forecasting company for clients who run offshore oilrigs.

There were two evaluation criteria; Clarity was explained as indicating how understandable a forecast was, and Readability as indicating how fluent and readable it was. The experiment involved 22 forecast dates and outputs from the 10 systems described in (Belz and Kow, 2009) (also included in the corpus release) for those dates (as well as the corresponding forecasts in the corpus) in the evaluation, i.e. a total of 242 forecast texts.

### 4.2 Rating scale experiment

We used the results of a previous experiment (Belz and Kow, 2009) in which participants were asked to rate forecast texts for Clarity and Readability, each on a scale of 1–7.

The 22 participants were all University of Brighton staff whose first language was English and who had no experience of NLP. While earlier experiments used master mariners as well as lay-people in a similar evaluation (Belz and Reiter, 2006), these experiments also demonstrated that the correlation between the ratings by expert evaluators and lay-people is very strong in the METEO domain (Pearson's  $r = 0.845$ ).

We used a single 22 (evaluators) by 22 (test data items) Latin Square; there were 484 trials in this experiment.

### 4.3 Preference judgement experiment

Our new experiment used our standardised preference strength sliders (bottom of Figure 2). We recruited 22 different evaluators from among students currently completing or recently having

Type	Measure	Clarity	Readability
RSE	$F_{(10,473)}$	23.507**	24.351**
	N sig diffs	24/55	23/55
	K's W	.497**	.533**
	(Text $F_{(21,462)}$ )	1.467	1.961**
	(Evaluator $F_{(21,462)}$ )	4.832**	4.824**
PJE	$F_{(10,1865)}$	45.081**	41.318**
	N sig diffs	34/55	32/55
	K's W	.626**	.542**
	(Text $F_{(21,916)}$ )	1.436	1.573
	(Evaluator $F_{(21,921)}$ )	.794	1.057

Table 3: METEO RSE/PJE: Results of analyses looking at effect of System.

completed a linguistics-related degree at Oxford, KCL, UCL, Sussex and Brighton.

We had at our disposal 11 METEO systems, so there were  $\binom{11}{2} = 55$  system combinations to evaluate on the 22 test data items. We decided on a design of ten  $11 \times 11$  Latin Squares to accommodate the 55 system pairings, so there was a total of 1210 trials in this experiment.

#### 4.4 Results and comparative analysis

Table 3 shows the same types of comparative measures as in the previous section. Note that the self-consistency measure is missing, because for METEO-PJE we do not have multiple scores for the same pair of systems by the same evaluator.

For the METEO task, the relative amount variation in Clarity and Radability accounted for by System is similar in the RSE, and again similar in the PJE. However, F ratios and numbers of significant differences found are higher in the latter than in the RSE. The inter-evaluator agreement measure also has higher values for both Clarity and Readability in the PJE, although the difference is much more pronounced in the case of Clarity.

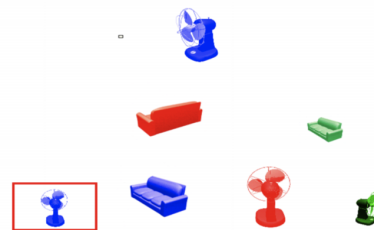
In the RSE version, Evaluator has a small but significant effect on both Clarity and Readability, which disappears in the PJE version. Similarly, a small effect of Text (date of weather forecast in this data set) on Fluency in the RSE version disappears in the PJE version.

### 5 RSE/PJE Pair 2: Descriptions of furniture items and faces

#### 5.1 Data and generic design

In our third pair of evaluations, we used the system outputs from the TUNA'09 shared-task com-

petition (Gatt et al., 2009).<sup>5</sup> The TUNA data is a collection of images of domain entities paired with descriptions of entities. Each pair consists of seven entity images where one is highlighted (by a red box surrounding it), paired with a description of the highlighted entity, e.g.:



the small blue fan

The descriptions were collected in an online experiment with anonymous participants, and then annotated for semantic content. In TUNA'09, the task for participating systems was to generate descriptions of the highlighted entities given semantic representations of all seven entities. In the evaluation experiments, evaluators were asked to give two ratings in answer to the following questions (the first for Adequacy, the second for Fluency):

1. How clear is this description? Try to imagine someone who could see the same grid with the same pictures, but didn't know which of the pictures was the target. How easily would they be able to find it, based on the phrase given?
2. How fluent is this description? Here your task is to judge how well the phrase reads. Is it good, clear English?

Participants were shown a system output, together with its corresponding domain, displayed as the set of corresponding images on the screen. The intended (target) referent was highlighted by a red frame surrounding it on the screen.

Following detailed instructions, subjects did two practice examples, followed by the 112 test items in random order.

There were 8 'systems' in the TUNA evaluations: the descriptions produced by the 6 systems and two sets of humans-authored descriptions.

#### 5.2 Rating scale experiment

The rating scale experiment that was part of the TUNA'09 evaluations had a design of fourteen  $8 \times 8$  squares, and a total of 896 trials.

<sup>5</sup>The TUNA'09 data and documentation is available for download from <http://www.nltg.brighton.ac.uk/home/Anja.Belz>

Type	Measure	Adequacy	Fluency
RSE	$F_{(7,888)}$	6.371**	17.207**
	N sig diffs	7/28	15/28
	K's W	.471**	.676**
	( Text1	$F_{(111,784)}$	1.519**
( Text2	$F_{(14,881)}$	8.992**	4.694** )
( Evaluator	$F_{(7,888)}$	13.136**	17.479** )
PJE	$F_{(7,6264)}$	46.503**	89.236**
	N sig diffs	19/28	22/28
	K's W	.573**	.654**
	( Text1	$F_{(111,3024)}$	.746
( Text2	$F_{(14,3121)}$	.856	.853 )
( Evaluator	$F_{(27,3108)}$	1.3	1.638* )

Table 4: TUNA RSE/PJE: Results of analyses looking at effect of System.

Subjects were asked to give their judgments for Clarity and Fluency for each item by manipulating a slider. The slider pointer was placed in the center at the beginning of each trial. The position of the slider selected by the subject mapped to an integer value between 1 and 100. However, the scale was not visible to participants who knew only that one end of the scale corresponded to the worst possible score and the opposite end to the best.

Eight native speakers of English were recruited for this experiment from among post-graduate students currently doing a Masters degree in a linguistics-related subject at UCL, Sussex and Brighton universities.

### 5.3 Preference judgement experiment

Our new experiment used our standardised preference strength sliders (bottom of Figure 2). To accommodate all pairwise comparisons as well as all test set items, we used a design of four  $28 \times 28$  Latin Squares, and recruited 28 evaluators from among students currently completing, or recently having completed, a degree in a linguistics-related subject at Oxford, KCL, UCL, Sussex and Brighton universities. There were 3,136 trials in this version of the experiment.

### 5.4 Results and comparative analysis

Table 4 shows the same measures as we reported for the other two experiment pairs above. The picture is somewhat similar in that the measures have better values for PJE version except for the inter-evaluator agreement (Kendall's W) for Fluency which is slightly higher for the RSE version.

For the TUNA dataset, we look at two Text factors. Text2 refers to different sets of entities used in trials; there are 15 different ones. Text1 refers to sets of entities and their specific distribution over the visual display grid in trials (see the figure in Section 5.1); there are 112 different combinations of entity set and grid locations.

The most striking aspect of the results in Table 4 is the effect of Evaluator in the RSE version which appears to account for more variability in the data even than System (relative to other factors). In fact, in the case of Adequacy, even Text2 causes more variation than System. In contrast, in the PJE version, by far the biggest cause of variability is System (for both criteria), and the F ratios for Text and Evaluators are not significant except for Evaluator on Fluency (weakly significant at .05).

On the face of it, the variation between evaluators in the RSE version as evidenced by the F ratio is worrying. However, Kendall's W shows that in terms of mean rank, evaluators actually agreed similarly well on Fluency in both RSE and PJE. The F measure is based on mean scores whereas W is based on mean score ranks, so there was more variation in the absolute scores than in the ranks.

The reason is likely to be connected to the way ratings were expressed by evaluators in the TUNA-RSE experiment: recall that evaluators had the task of moving the pointer to the place on the slider bar that they felt corresponded to the quality of text being evaluated. As no numbers were visible, the only information evaluators had to go on was which was the 'worse' end and which was the 'better' end of the slider. It seems that different evaluators used this evaluation tool in very different ways (accounting for the variation in absolute scores), but were able to apply their way of using the tool reasonably consistently to different texts (so that they were able to achieve reasonably good agreement with the other evaluators in terms of relative scores).

## 6 Discussion

We have looked at a range of aspects of evaluation experiments: the effect of the factors System, Text and Evaluator on evaluation scores; the number of significant differences between systems found; self-consistency; and inter-evaluator agreement (as described by F ratios obtained in one-way ANOVAs for Evaluator, as well as by Kendall's W measuring inter-evaluator agreement).

The results are unambiguous as far as the Clarity criterion (called Adequacy in TUNA) is concerned: in all three experiment pairs, the preference-strength judgement (PSE) version had a greater effect of System, a smaller effect of Text and Evaluator, more significant pairwise differences, better inter-evaluator agreement, and (where we were able to measure it) better self-consistency.

The same is true for Readability in METEO and Fluency in TUNA, in the latter case except for W which is slightly lower in TUNA-PJE than TUNA-RSE. However, Readability in GREC-NEG bucks the trend: here, all measures are worse in the PJE version than in the RSE version (although for the W measures, the differences are small). Part of the reason for this may be that in GREC-NEG PJE each system was only compared to one single other ‘system’, the (human-authored) original Wikipedia texts.

If we see less effect of Clarity than of Fluency in an experiment (as in GREC-NEG RSE and TUNA RSE), then we might want to conclude that systems differed less in terms of Clarity than in terms of Fluency. However, the real explanation may be that evaluators simply found it harder to apply the Clarity criterion than the Fluency criterion in a given evaluation set-up. The fact that the difference in effect between Fluency and Clarity virtually disappears in GREC-NEG PJE makes this the more likely explanation at least for the GREC-NEG evaluations.

Parametric statistics are more powerful than non-parametric ones because of the strong assumptions they make about the nature of the data. Roughly speaking, they are more likely to uncover significant differences. Where the assumptions are violated, the risk is that significance is overestimated (the likelihood that null hypotheses are incorrectly rejected increases). One might consider using a slider mapping to a continuous scale instead of a multiple-choice rating form in order to overcome this problem, but the evidence from the TUNA RSE evaluation appears to be that this can result in unacceptably large variation in how individual evaluators apply the scale to assign absolute scores.

What seems to make the difference in terms of ease of application of evaluation criteria and reduction of undesirable effects is not the use of continuous scales (as e.g. implemented in slider bars),

but the comparative element, where pairs of systems are compared and one is selected as better in terms of a given criterion than the other.

It makes sense intuitively that deciding which of two texts is clearer should be an easier task than deciding whether a system is a 5, 4, 3 or 1 in terms of its clarity. PJE enabled evaluators to apply the Clarity criterion to determine ranks more consistently in all three experiment pairs.

However, it was an open question whether evaluators would also be able to express the *strength* of their preference consistently. From the results we report here it seems clear that this is indeed the case: the System F ratios which look at absolute scores (in the PJE quantifying the strength of a preference) are higher, and the Evaluator F ratios lower, in all but one of the experiments.

While there were the same number of trials in the two GREC-NEG evaluations, there were 2.5 times as many trials in METEO-PJE than in METEO-RSE, and 3.5 times as many trials in TUNA-PJE than in TUNA-RSE. The increase in trials is counter-balanced to some extent by the fact that evaluators tend to give relative judgements far more quickly than absolute judgements, but clearly there is an increase in cost associated with including all system pairings in a PJE. If this cost grows unacceptably large, a subset of systems has to be selected as reference systems.

## 7 Concluding Remarks

Our aim in the research presented in this paper was to investigate how rating-scale experiments compare to preference-strength judgement experiments in the evaluation of automatically generated language. We find that preference-strength judgement evaluations generally have a greater relative effect of System (the factor actually under investigation), a smaller relative effect of Text and Evaluator (whose effect should be small), a larger number of significant pairwise differences between systems, better inter-evaluator agreement, and (where we were able to measure it) better evaluator self-consistency.

## References

- Anja Belz and Eric Kow. 2009. System building cost vs. output quality in data-to-text generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 16–24.

- A. Belz and E. Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceedings of EACL'06*, pages 313–320.
- Anja Belz, Eric Kow, and Jette Viethen. 2009. The GREC named entity generation challenge 2009: Overview and evaluation results. In *Proceedings of the ACL-IJCNLP'09 Workshop on Language Generation and Summarisation (UCNLG+Sum)*, pages 88–98.
- A. Belz. 2009. Prodigy-METEO: Pre-alpha release notes (Nov 2009). Technical Report NLTG-09-01, Natural Language Technology Group, CMIS, University of Brighton.
- Bernard Choi and Anita Pak. 2005. A catalog of biases in questionnaires. *Preventing Chronic Disease*, 2(1):A13.
- A. Gatt, A. Belz, and E. Kow. 2009. The TUNA Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG'09)*, pages 198–206.
- C.-Y. Lin and F. J. Och. 2004. ORANGE: A method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, pages 501–507, Geneva.
- E. Reiter, S. Sripada, J. Hunter, and J. Yu. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137–169.
- Sidney Siegel. 1957. Non-parametric statistics. *The American Statistician*, 11(3):13–19.
- S. Sripada, E. Reiter, J. Hunter, and J. Yu. 2002. SUMTIME-METEO: A parallel corpus of naturally occurring forecast texts and weather data. Technical Report AUCS/TR0201, Computing Science Department, University of Aberdeen.
- H. Trang Dang. 2006. DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the COLING-ACL'06 Workshop on Task-Focused Summarization and Question Answering*, pages 48–55.
- J. Turian, L. Shen, and I. D. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of MT Summit IX*, pages 386–393, New Orleans.





# A Discourse-Aware Graph-Based Content-Selection Framework

Seniz Demir Sandra Carberry Kathleen F. McCoy

Department of Computer Science

University of Delaware

Newark, DE 19716

{demir, carberry, mccoy}@cis.udel.edu

## Abstract

This paper presents an easy-to-adapt, discourse-aware framework that can be utilized as the content selection component of a generation system whose goal is to deliver descriptive texts in several turns. Our framework involves a novel use of a graph-based ranking algorithm, to iteratively determine what content to convey to a given request while taking into account various considerations such as capturing a priori importance of information, conveying related information, avoiding redundancy, and incorporating the effects of discourse history. We illustrate and evaluate this framework in an accessibility system for sight-impaired individuals.

## 1 Introduction

Content selection is the task responsible for determining what to convey in the output of a generation system at the current exchange (Reiter and Dale, 1997). This very domain dependent task is extremely important from the perspective of users (Sripada et al., 2001) who have been observed to be tolerant of realization problems as long as the appropriate content is expressed. The NLG community has proposed various content selection approaches since early systems (Moore and Paris, 1993; McKeown, 1985) which placed emphasis on text structure and adapted planning techniques or schemas to meet discourse goals.

This paper proposes a domain-independent framework which can be incorporated as a content selection component in a system whose goal is to deliver descriptive or explanatory texts, such as the ILEX (O'Donnell et al., 2001), KNIGHT (Lester and Porter, 1997), and POLIBOX (Chiarcos and Stede, 2004) systems. At the core of our framework lies a novel use of a graph-based ranking al-

gorithm, which exploits discourse related considerations in determining what content to convey in response to a request for information. This framework provides the ability to generate successive history-aware texts and the flexibility to generate different texts with different parameter settings.

One discourse consideration is the tenet that the propositions selected for inclusion in a text should be in some way related to one another. Thus, the selection process should be influenced by the relevance of information to what has already been selected for inclusion. Moreover, we argue that if the information given in a proposition can be deduced from the information provided by any other proposition in the text, this would introduce redundancy and should be avoided.

Many systems (such as MATCH (Walker et al., 2004) and GEA (Carenini and Moore, 2006)) contain a user model which is employed to adapt content selection to the user's preferences (Reiter and Dale, 1997). Our framework provides a facility to model a stereotypical user by incorporating the a priori importance of propositions. This facility can also be used to capture the preferences of a particular user.

In a dialogue system, utterances that are generated without exploiting the previous discourse seem awkward and unnatural (Moore, 1993). Our framework takes the previous discourse into account so as to omit recently communicated propositions and to determine when repetition of a previously communicated proposition is appropriate.

To our knowledge, our work is the first effort utilizing a graph-based ranking algorithm for content selection, while taking into account what information preferably should and shouldn't be conveyed together, the a priori importance of information, and the discourse history. Our framework is a domain-independent methodology containing domain-dependent features that must be instantiated when applying the methodology to a domain.

Section 2 describes our domain-independent methodology for determining the content of a response. Section 3 illustrates its application in an accessibility system for sight-impaired individuals and shows the generation flexibility provided by this framework. Finally, Section 4 discusses the results of user studies conducted to evaluate the effectiveness of our methodology.

## 2 A Graph-based Content Selection Framework

Our domain-independent framework can be applied to any domain where there is a set of propositions that *might* be conveyed and where a bottom-up strategy for content selection is appropriate. It is particularly useful when the set of propositions should be delivered a little at a time. For example, the ILEX system (O’Donnell et al., 2001) uses multiple descriptions to convey the available information about a museum artifact, since the length of the text that can be displayed on a page is limited. In order to use our framework, an application developer should identify the set of propositions that might be conveyed in the domain, specify the relations between these propositions, and optionally assess a priori importance of the propositions.

Our framework uses a weighted undirected graph (**relation graph**), where the propositions are captured as vertices of the graph and the edges represent relations between these propositions. While the number and kinds of relations represented is up to the developer, the framework does require the use of one specific relation (**Redundancy Relation**) that is generalizable to any descriptive domain. Redundancy\_Relation must be specified between two propositions if they provide similar kinds of information or the information provided by one of the propositions can be deduced from the information provided by the other. For example, consider applying the framework to the ILEX domain. Since the proposition that “this jewelry is produced by a single craftsman” can be deduced from the proposition that “this jewelry is made by a British designer”, these propositions should be connected with a Redundancy\_Relation in the relation\_graph.

There is at most one edge between any two vertices and the weight of that edge represents how important it is to convey the corresponding propositions in the same text (which we refer to as the strength of the relation between these proposi-

tions). For example, suppose that once a museum artifact is introduced in ILEX, it is more important to convey its design style in the same description as opposed to where it is produced. In this case, the weight of the edge between the propositions introducing the artifact and its style should be higher than the weight of the edge between the propositions introducing the artifact and its production place.

The framework incorporates a stereotypical user model via an additional vertex (**priority\_vertex**) in the relation\_graph. The priority\_vertex is connected to all other vertices in the graph. The weight of the edge between a vertex and the priority\_vertex represents the a priori importance of that vertex, which in turn specifies the importance of the corresponding proposition. For example, suppose that in the ILEX domain an artifact has two features that are connected to the proposition introducing the artifact by the “feature-of” relation. The a priori importance of one of these features over the other can be specified by giving a higher weight to the edge connecting this proposition to the priority\_vertex than is given to the edge between the other feature and the priority\_vertex. This captures a priori importance and makes it more likely that the important feature will be included in the artifact’s description.

### 2.1 Our Ranking Algorithm

With this graph-based setting, the most important thing to say is the proposition which is most central. Several centrality algorithms have been proposed in the literature (Freeman, 1979; Navigli and Lapata, 2007) for calculating the importance scores of vertices in a graph. The well-known PageRank centrality (Brin and Page, 1998) calculates the importance of a vertex by taking into account the importance of all other vertices and the relation of vertices to one another. This metric has been applied to various tasks such as word sense disambiguation (Sinha and Mihalcea, 2007) and text summarization (Erkan and Radev, 2004). We adopted the weighted PageRank metric (Sinha and Mihalcea, 2007) for our framework and therefore compute the importance score of a vertex ( $V_x$ ) as:

$$PR(V_x) = (1 - d) + d * \sum_{(V_x, V_y) \in E} \frac{w_{yx}}{\sum_{(V_z, V_y) \in E} w_{yz}} PR(V_y)$$

where  $w_{xy}$  is the weight associated with the edge between vertices ( $V_x$ ) and ( $V_y$ ),  $E$  is the set of all

edges, and  $d$  is the damping factor, set to 0.85, which is its usual setting.

Once the propositions in a domain are captured in a `relation_graph` with weights assigned to the edges between them, the straightforward way of identifying the propositions to be conveyed in the generated text would be to calculate the importance of each vertex via the formula above and then select the  $k$  vertices with the highest scores. However, this straightforward application would fail to address the discourse issues cited earlier. Thus we select propositions incrementally, where with each proposition selected, weights in the graph are adjusted causing related propositions to be highlighted and redundant information to be repelled. Because our responses are delivered over several turns, we also adjust weights between responses to reflect that discourse situation.

Our algorithm, shown in Figure 1, is run each time a response text is to be generated. For each new response, the algorithm begins by adjusting the importance of the `priority_vertex` (making it high) and clearing the list of selected propositions. Step 2 is the heart of the algorithm for generating a single response. It incrementally selects propositions to include in the current response, and adjusts weights to reflect what has been selected. In particular, in order to select a proposition, importance scores are computed using the weighted PageRank metric for all vertices corresponding to propositions that have not yet been selected for inclusion in this response (Step 2-a), and only the proposition that receives the highest score is selected (Step 2-b). Then, adjustments are made to achieve four goals toward taking discourse information into account (Steps 2-c thru 2-g) before the PageRank algorithm is run again to select the next proposition. Steps 3 and 4 adjust weights to reflect the completed response and to prepare for generating the next response.

Our first goal is to reflect the a priori importance of propositions in the selection process. For this purpose, we always assign the highest (or one of the highest) importance scores to the `priority_vertex` among the other vertices (Steps 1 and 2-g). This will make the `priority_vertex` as influential as any other neighbor of a vertex when calculating its importance.

Our second goal is to select propositions that are relevant to previously selected propositions, or in terms of the graph-based notation, to **attract** the

selection of vertices that are connected to the selected vertices. To achieve this, we increase the importance of the vertices corresponding to selected propositions so that the propositions related to them have a higher probability of being chosen as the next proposition to include (Step 2-g).

Our third goal is to avoid selecting propositions that preferably shouldn't be communicated with previously selected propositions if other related propositions are available. To accomplish this, we introduce the term **repellers** to refer to the kinds of relations between propositions that are dispreferred over other relations once one of the propositions is selected for inclusion. Once a proposition is selected, we penalize the weights on the edges between the corresponding vertex and other vertices that are connected by a repeller (Step 2-d). We don't provide any general repellers in the framework, but rather this is left for the developer familiar with the domain; any number (zero or more) and kinds of relations could be identified as repellers for a particular application domain. For example, suppose that in the ILEX domain, some artifacts (such as necklaces) have as features both a set of design characteristics and the person who found the artifact. Once the artifact is introduced, it becomes more important to present the design characteristics rather than the person who found that artifact. This preference might be captured by classifying the relation connecting the proposition conveying the person who found it to the proposition introducing the artifact as a repeller.

Our fourth goal is to avoid redundancy by discouraging the selection of propositions connected by a `Redundancy_Relation` to previously selected propositions. Once a proposition is selected, we identify the vertices (**redundant\_to\_selected vertices**) which are connected to the selected vertex by the `Redundancy_Relation` (Step 2-e). For each `redundant_to_selected` vertex, we penalize the weights on the edges of the vertex except the edge connected to the `priority_vertex` (Step 2-f) and hence decrease the probability of that vertex being chosen for inclusion in the same response.

We have so far described how the content of a single response is constructed in our framework. To capture a situation where the system is engaged in a dialogue with the user and must generate additional responses for each subsequent user request, we need to ensure that discourse flows naturally. Thus, the ranking algorithm must take the previ-

1. Set an importance score to the **priority\_vertex** and empty the **selected\_vertices** set
2. Repeat steps 2-a to 2-g until the stopping criteria is met:
  - (a) use the weighted PageRank metric to calculate the importance scores of all vertices excluding the **priority\_vertex** and the vertices in the **selected\_vertices** set
  - (b) mark the vertex that received the highest score as **selected** and add it to the **selected\_vertices** set
  - (c) decrease the weight of the edge between the **selected\_vertex** and the **priority\_vertex**
  - (d) penalize the weights of the edges between the **selected\_vertex** and the vertices which are connected to it by a repeller via the penalty factor, if they weren't already adjusted
  - (e) find the vertices that are connected by a Redundancy\_Relation to the **selected\_vertex** (if any) and mark them as **redundant\_to\_selected**
  - (f) penalize the weights of all edges of **redundant\_to\_selected** vertices via the redundancy penalty factor except the edges connected to the **priority\_vertex**
  - (g) set the importance scores of the **priority\_vertex** and the vertices in the **selected\_vertices** set to the highest importance score calculated so far
3. Penalize the weights of the edges of the vertices in the **selected\_vertices** set via the penalty factor, if they weren't adjusted in Step 2
4. Increase the weights of the edges of all other vertices (excluding the vertices in the **selected\_vertices** set and **redundant\_to\_selected** vertices) via the boost factor

Figure 1: Our Ranking Algorithm for Content Selection.

ous discourse into account in order to identify and preferably select propositions that have not been conveyed before and to determine when repetition of a previously communicated proposition is appropriate. So once a proposition is included in a response, we have to reduce its ability to compete for inclusion in subsequent responses. Thus once a proposition is conveyed in a response, the weight of the edge connecting the corresponding vertex to the **priority\_vertex** is reduced (Step 2-c in Figure 1). Once a response is completed, we penalize the weights of the edges of each vertex that has been selected for inclusion in the current response via a penalty factor (if they aren't already adjusted) (Step 3 in Figure 1). We use the same penalty factor (which is used in Step 2-d in Figure 1) on each edge so that all edges connected to a selected vertex are penalized equally. However, it isn't enough just to penalize the edges of the vertices corresponding to the communicated propositions. Even after the penalties are applied, a proposition that has just been communicated might receive a higher importance score than an uncommunicated proposition<sup>1</sup>. In order to allow all propositions to become important enough to be said at some point, the algorithm increases the weights of the edges of all other vertices in the graph if they haven't already been decreased (Step 4 in Figure 1), thereby increasing their ability to compete in subsequent responses. In the current implementation, the weight of an edge is increased via a boost factor after a response if it is not connected to a proposition included in that response. The

<sup>1</sup>We observed that it might happen if a vertex is connected only to the **priority\_vertex**.

boost factor ensures that all propositions will eventually become important enough for inclusion.

### 3 Application in a Particular Domain

This section illustrates the application of our framework to a particular domain and how our framework facilitates flexible content selection. Our example is content selection in the SIGHT system (Elzer et al., 2007), whose goal is to provide visually impaired users with the knowledge that one would gain from viewing information graphics (such as bar charts) that appear in popular media. In the current implementation, SIGHT constructs a brief initial summary (Demir et al., 2008) that conveys the primary message of a bar chart along with its salient features. We enhanced the current SIGHT system to respond to user's follow-up requests for more information about the graphic, where the request does not specify the kind of information that is desired.

The first step in using our framework is determining the set of propositions that might be conveyed in this domain. In our earlier work (Demir et al., 2008), we identified a set of propositions that capture information that could be determined by looking at a bar chart, and for each message type defined in SIGHT, specified a subset of these propositions that are related to this message type. In our example, we use these propositions as candidates for inclusion in follow-up responses. Figure 2 presents a portion of the relation\_graph, where some of the identified propositions are represented as vertices.

The second step is optionally assessing the a priori importance of each proposition. In user

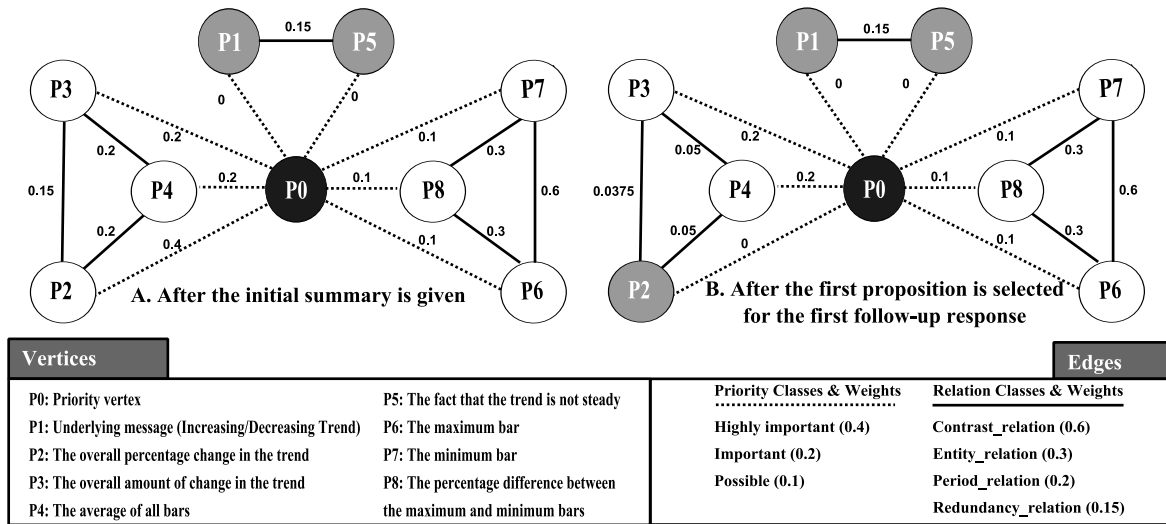


Figure 2: Subgraph of the Relation\_graph for Increasing and Decreasing Trend Message Types.

studies (Demir et al., 2008), we asked subjects to classify the propositions given for a message type into one of three classes according to their importance for inclusion in the initial summary: **essential**, **possible**, and **not important**. We leverage this information as the a priori importance of vertices in our graph representation. We define three priority classes. For the propositions that were not selected as *essential* by any participant, we classify the edges connecting these propositions to the priority\_vertex into **Possible** class. For the propositions which were selected as *essential* by a single participant, we classify the edges connecting them to the priority\_vertex into **Important** class. The edges of the remaining propositions are classified into **Highly Important** class. In this example instantiation, we assigned different numeric scores to these classes where Highly\_Important and Possible received the highest and lowest scores respectively.

The third step requires specifying the relations between every pair of related propositions and determining the weights associated with these relations in the relation\_graph. First, we identified propositions which we decided should be connected by the Redundancy\_Relation (such as the propositions conveying “the overall amount of change in the trend” and “the range of the trend”). Next, we had to determine other relations and assign relative weights. Instead of defining a unique relation for each related pair, we defined three relation classes, and assigned the relations between related propositions to one of these classes:

- **Period\_Relation:** expresses a relation between two propositions that span the same time period

- **Entity\_Relation:** expresses a relation between two propositions if the entities involved in the propositions overlap
- **Contrast\_Relation:** expresses a relation between two propositions if the information provided by one of the propositions contrasts with the information provided by the other

We determined that it was very common in this domain to deliver contrasting propositions together (similar to other domains (Marcu, 1998)) and therefore we assigned the highest score to the Contrast\_Relation class. For local focusing purposes, it is desirable that propositions involving common entities be delivered in the same response and thus the Entity\_Relation class was given the second highest score. On the other hand, two propositions which only share the same period are not very related and conveying such propositions in the same response could cause the text to appear “choppy”. We thus identified the Period\_Relation class as a repeller and assigned the second lowest score to relations in that class. Since we don’t want redundancy in the generated text, the lowest score was assigned to the Redundancy\_Relation class. The next section shows how associating particular weights with the priority and relation classes changes the behavior of the framework.

In the domain of graphics, a collection of descriptions of the targeted kind which would facilitate a learning based model isn’t available. However, the accessibility of a corpus in a new domain would allow the identification of the propositions along with their relations to each other and the determination of what weighting scheme and adjustment policy will produce the corpus within reasonable bounds.

### 3.1 Generating Flexible Responses

The behavior of our framework is dependent on a number of design parameters such as the weights associated with various relations, the identification of repellers, the a priori importance of information (if applicable), and the extent to which conveying redundant information should be avoided. The framework allows the application developer to adjust these factors resulting in the selection of different content and the generation of different responses. For instance, in a very straightforward setting where the same numeric score is assigned to all relations, the a priori importance of information would be the major determining factor in the selection process. In this section, we will illustrate our framework's behavior in SIGHT with three different scenarios. In each case, the user is assumed to post two consecutive requests for additional information about the graphic in Figure 3 after receiving its initial summary.

In our first scenario (which we refer to as “base-setting”), the following values have been given to various design parameters that must be specified in order to run the ranking algorithm. 1) The weights of the relations are set to the numeric scores shown in the text labelled **Edges** at the bottom (right side) of Figure 2. 2) The stopping criteria which specifies the number of propositions selected for inclusion in a follow-up response (Step 2 in Figure 1) is set to four. 3) The amount of decrease in the weight of the edge between the `priority_vertex` and the vertex selected for inclusion (Step 2-c in Figure 1) is set to that edge's original weight. Thus, in our example, the weight of that edge is set to 0 once a proposition has been selected for inclusion. 4) The penalty and the redundancy penalty factors which are used to penalize the edges of a selected vertex and the vertices redundant to the selected vertex (Steps 2-d and 3, and 2-f in Figure 1) are set to the quotient of the highest numeric score initially assigned to a relation class divided by the lowest numeric score initially assigned to a relation class. A penalized score for a relation class is computed by dividing its initial score by the penalty factor. The edges of a vertex are penalized by assigning the penalized scores to these edges based on the relations that they represent. This setting guarantees that the weight of an edge which represents the strongest relation cannot be penalized to be lower than the score initially assigned to the weakest relation. 5) The boost factor which

is used to favor the selection of previously unconveyed propositions for inclusion in subsequent responses (Step 4 in Figure 1) is set to the square root of the penalty factor. Thus, the weights of the edges connected to vertices of previously communicated propositions are restored to their initial scores slowly.

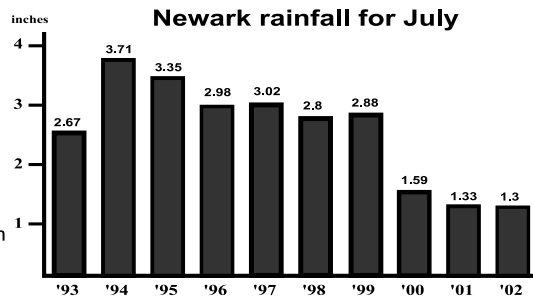
Since in our example, the initial summary has already been presented, we treat the propositions conveyed in that summary (P1 and P5 in Figure 2) as if they had been conveyed in a follow-up response and penalize the edges of their corresponding vertices (Steps 2-c and 3 in Figure 1). Thus, before we invoke the algorithm to construct the first follow-up response, the weights of edges of the graph are as shown in Figure 2-A. Within this base-setting, SIGHT generates the set of follow-up responses shown in Figure 3A.

In our first scenario (base-setting), we assumed that the user is capable of making mathematical deductions such as inferring “the overall amount of change in the trend” from “the range of the trend”; thus we identified such propositions as sharing a `Redundancy_Relation`. Young readers (such as fourth graders) might not find these propositions as redundant because they are lacking in mathematical skills. In our second scenario, we address this issue by setting the redundancy penalty factor to 1 (Step 2-f in Figure 1) and thus eliminate the penalty on the `Redundancy_Relation`. Now, for the same graphic, SIGHT generates, in turn, the second alternative set of responses shown in Figure 3B. The responses for the two scenarios differ in the second follow-up response. In the first scenario, a description of the smallest drop was included. However, in the second scenario, this proposition is replaced with the overall amount of change in the trend. This proposition was excluded in the first scenario because the redundancy penalty factor made it drop in importance.

Our third scenario shows how altering the weights assigned to relations may change the responses. Consider a situation where the `Contrast_Relation` is given even higher importance by doubling its score; this might occur in a university course domain where courses on the same general topic are contrasted. SIGHT would then generate the third alternative set of follow-up responses shown in Figure 3C. The algorithm is more strongly forced to group propositions that

**Initial Summary**

Following a moderate rise between the year 1993 and the year 1994, the graphic shows a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The amount of newark rainfall for july shows the largest drop of about 1.29 inches between the year 1999 and the year 2000. With the exception of a few rises, slight decreases are observed almost every year over the period from the year 1994 to the year 2002.

**First follow-up response:**

The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. The amount of this rainfall for july averages 2.55 inches.

**Second follow-up response:**

Recall that there is a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The amount of newark rainfall for july shows the smallest drop of about 0.03 inches between the year 2001 and the year 2002. The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july.

**A. First alternative set of responses is shown above (base-setting)****First follow-up response:**

The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. The amount of this rainfall for july averages 2.55 inches.

**Second follow-up response:**

Recall that there is a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july. **The difference between the amount of newark rainfall for july in the year 1994 and that in the year 2002 is 2.41 inches.**

**B. Second alternative set of responses is shown above (the Redundancy\_Relation is not penalized)****First follow-up response:**

The amount of newark rainfall for july ranges from 3.71 to 1.3 inches and shows a decrease of nearly 65 percent over the period from the year 1994 to the year 2002. **The year 1994 at 3.71 inches has the highest amount of rainfall for july and the year 2002 at 1.3 inches has the lowest amount of rainfall for july.**

**Second follow-up response:**

Recall that there is a decreasing trend over the period from the year 1994 to the year 2002 in the amount of newark rainfall for july, which **shows the largest drop of 1.29 inches between the year 1999 and the year 2000**. At the year 1997 and the year 1999, unusual rises are observed in the amount of this rainfall for july, which **shows the smallest drop of 0.03 inches between the year 2001 and the year 2002**.

**C. Third alternative set of responses is shown above (the numeric score of the Contrast\_Relation is doubled)**

Figure 3: Initial Summary and Follow-up Responses.

are in a contrast relation (shown in bold), which changes the ranking of these propositions.

## 4 Evaluation

To determine whether our framework selects appropriate content within the context of an application, and to assess the contribution of the discourse related considerations to the selected content and their impact on readers' satisfaction, we conducted two user studies. In both studies, the participants were told that the initial summary should include the most important information about the graphic and that the remaining pieces of information should be conveyed via follow-up responses. The participants were also told that the information in the first response should be more important than the information in subsequent responses.

Our goal in the first study was to evaluate the effectiveness of our framework (base-setting) in determining the content of follow-up responses in SIGHT. To our knowledge, no one else has gener-

ated high-level descriptions of information graphics, and therefore evaluation using implementations of existing content selection modules in the domain of graphics as a baseline is not feasible. Thus, we evaluated our framework by comparing the content that it selects for inclusion in a follow-up response for a particular graphic with the content chosen by human subjects for the same response. Twenty one university students participated in the first study and each participant was presented with the same four graphics. For each graphic, the participants were first presented with its initial summary and the set of propositions (18 different propositions) that were used to construct the relation\_graph in our framework. The participants were then asked to select the four propositions that they thought were most important to convey in the first follow-up response.

For each graphic, we ranked the propositions with respect to the number of times that they were selected by the participants and determined the position of each proposition selected by our frame-

work for inclusion in the first follow-up response with respect to this ranking. The propositions selected by our framework were ranked by the participants as the *1st*, *2nd*, *3rd*, and *5th* in the first graphic, as the *1st*, *3rd*, *4th*, and *5th* in the second graphic, as the *1st*, *2nd*, *3rd*, and *6th* in the third graphic, and as the *2nd*, *3rd*, *4th*, and *6th* in the fourth graphic. Thus for every graph, three of the four propositions selected by our framework were also in the top four highly-rated propositions selected by the participants. Therefore, this study demonstrated that our content selection framework selects the most important information for inclusion in a response at the current exchange.

We argued that simply running PageRank to select the highly-rated propositions is likely to lead to text that does not cohere because it may contain unrelated or redundant propositions, or fail to communicate related propositions. Thus, our approach iteratively runs PageRank and includes discourse related factors in order to allow what has been selected to influence the future selections and consequently improve text coherence. To verify this argument, we conducted a second study with four graphics and two different sets of follow-up responses (each consisting of two consecutive responses) generated for each graphic. We constructed the first set of responses (**baseline**) by running PageRank to completion and selecting the top eight highly-rated propositions, where the top four propositions form the first response. The content of the second set of responses was identified by our approach. Twelve university students (who did not participate in the first study) were presented with these four graphics along with their initial summaries. Each participant was also presented with the set of responses generated by our approach in two graphics and the set of responses generated by the baseline in other cases; the participants were unaware of how the follow-up responses were generated. Overall, each set of responses was presented to six participants.

We asked the participants to evaluate the set of responses in terms of their quality in conveying additional information (from 1 to 5 with 5 being the best). We also asked each participant to choose which set of responses (from among the four sets of responses presented to them) best provides further information about the corresponding graphic. The participants gave the set of responses generated by our approach an average rat-

ing of **4.33**. The average participant rating for the set of responses generated by the baseline was **3.96**. In addition, the lowest score given to the set of responses generated by our approach was 3, whereas the lowest score that the baseline received was 2. We also observed that the set of responses generated by our approach was selected as the best set by eight of the twelve participants. Three of the remaining four participants selected the set of responses generated by the baseline as best (although they gave the same score to a set of responses generated by our approach). In these cases, the participants emphasized the wording of the responses as the reason for their selection. Thus this study demonstrated that the inclusion of discourse related factors in our approach, in addition to the use of PageRank (which utilizes the a priori importance of the propositions and their relations to each other), contributes to text coherence and improves readers' satisfaction.

## 5 Conclusion

This paper has presented our implemented domain-independent content selection framework, which contains domain-dependent features that must be instantiated when applying it to a particular domain. To our knowledge, our work is the first to select appropriate content by using an incremental graph-based ranking algorithm that takes into account the tendency for some information to seem related or redundant to other information, the a priori importance of information, and what has already been said in the previous discourse. Although our framework requires a knowledge engineering phase to port it to a new domain, it handles discourse issues without requiring that the developer write code to address them. We have demonstrated how our framework was incorporated in an accessibility system whose goal is the generation of texts to describe information graphics. The evaluation studies of our framework within that accessibility system show its effectiveness in determining the content of follow-up responses.

## 6 Acknowledgements

The authors would like to thank Debra Yarrington and the members of the NLP-AI Lab at UD for their help throughout the evaluation of this work. This material is based upon work supported by the National Institute on Disability and Rehabilitation Research under Grant No. H133G080047.



## References

- S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.
- G. Carenini and J. Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–452.
- C. Chiarcos and M. Stede. 2004. Saliency-Driven Text Planning. *In Proc. of INLG'04*.
- S. Demir, S. Carberry, and K. F. McCoy. 2008. Generating Textual Summaries of Bar Charts. *In Proc. of INLG'08*.
- S. Elzer, E. Schwartz, S. Carberry, D. Chester, S. Demir, and P. Wu. 2007. A browser extension for providing visually impaired users access to the content of bar charts on the web. *In Proc. of WEBIST'2007*.
- G. Erkan and D. Radev. 2004. LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.
- L. C. Freeman. 1979. Centrality in Social Networks: I. Conceptual Clarification. *Social Networks*, 1:215–239.
- J. Lester and B. Porter. 1997. Developing and empirically evaluating robust explanation generators: the KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.
- D. Marcu. 1998. The rhetorical parsing, summarization, and generation of natural language texts. *PhD. Thesis, Department of Computer Science, University of Toronto*.
- K. McKeown. 1985. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41.
- J. Moore and C. Paris. 1993. Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694.
- J. Moore. 1993. Indexing and exploiting a discourse history to generate context-sensitive explanations. *In Proc. of HLT'93*, 165–170.
- R. Navigli and M. Lapata. 2007. Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. *In Proc. of IJCAI'07*, 1683–1688.
- M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. 2001. ILEX: an architecture for a dynamic hypertext generation system. *In Natural Language Engineering*, 7(3):225–250.
- E. Reiter and R. Dale. 1997. Building applied natural language generation systems. *In Natural Language Engineering*, 3(1):57–87.
- R. Sinha and R. Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. *In Proc. of ICSC'07*.
- S. Sripada, E. Reiter, J. Hunter, and J. Yu. 2001. A Two-Stage Model for Content Determination. *In Proc. of ENLWG'01*.
- M. Walker, S.J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *In Cognitive Science*, 28(5):811–840.



# Generating Referring Expressions with Reference Domain Theory

Alexandre Denis

TALARIS team / UMR 7503 LORIA/INRIA

Lorraine. Campus scientifique, BP 239

F-54506 Vandoeuvre-lès-Nancy cedex

alexandre.denis@loria.fr

## Abstract

In this paper we present a reference generation model based on Reference Domain Theory which gives a dynamic account of reference. This reference model assumes that each referring act both relies and updates the reference context. We present a formal definition of a reference domain, a generation algorithm and its instantiation in the GIVE challenge.

a reference domain but also a reference domain in a set of reference domains that we call here *referential space*. Moreover each referring act presupposes a given state of the referential space, and the explicit representation of these presuppositions as constraints on the suitable domain for interpretation or generation allows the implementation of a reversible reference module. We will focus here on generation. Details about the interpretation side of RDT can be found in (Salmon-Alt and Romary, 2001; Denis et al., 2006).

## 1 Introduction

Reference is a process in which participants interpret and produce their referring expressions according to the previous context. But as Stalnaker puts it: the discourse context “is both the object on which speech acts and the source of the information relative to which speech acts are interpreted” (Stalnaker, 1998). To put it briefly, referring acts not only rely on the context to produce a reference but also *modify* it. This aspect is not taken into account in the classical generation algorithm by (Dale and Reiter, 1995). Each referent is generated by discriminating it inside a context. However, the construction and update of this context is not addressed.

Further literature on reference generation partially gives an account for the dynamic nature of the referring process. For example in (Krahmer and Theune, 2002), each referring act increases the salience of the referent such that further references can be made according to a smaller context, namely the set of objects whose salience is greater than the referent’s salience. Reference Domain Theory (RDT) (Reboul, 1998; Salmon-Alt and Romary, 2001) goes a step further by assuming that referring acts make salient the context sets *themselves*. This theory addresses the construction and update of the context sets, called in this theory *reference domains*. The goal of a referring act is then to discriminate a referent inside

However most of the previous work on RDT does not address computational details. Although (Salmon-Alt and Romary, 2000) provides a generation algorithm, the formal definition of a reference domain and the explicit representation of the constraints are not provided. In this paper we show how RDT can be used to generate referring expressions. The context of our work is the GIVE challenge (Byron et al., 2007; Byron et al., 2009). This challenge aims to evaluate instruction generation systems in a situated setting. The goal is to provide instructions to a player in a 3D maze in order to guide him to find a hidden trophy. We are here interested with the referring aspect involved in GIVE: the player has to push buttons to open doors or disable alarms, thus the system has to generate referring expressions to these buttons.

We first present in section 2 some definitions, then in section 3 we detail a generic generation algorithm. Section 4 shows a use case of RDT in the context of the GIVE challenge and provides a detailed example of the reference process. The presented model is generic, but all the examples given throughout the paper refer to the GIVE setting. Eventually, in section 6 we conclude the paper by demonstrating the success of RDT in an evaluation based on the GIVE setting.

## 2 Definitions

The referring process is a *discrimination* process whose goal is to discriminate one or more individuals in a context set. The discrimination can make use of different sources of information. It can be a *semantic* discrimination, for instance by uttering semantic properties possessed by the referent to rule out distractors, e.g. “the blue button”. It can be a discrimination of the *focus*, that is to make use of the current center of attention, e.g. “this button” or “the other button”. The discrimination can also rely on the previous referring acts, for instance when uttering “Push a blue button. Yes this one”, where “this one” would be unambiguously uttered in a context of a red and a blue button thanks to the mention of “a blue button”. A reference model has to take into account these different ways to discriminate.

On the other hand, a reference model has also to consider how objects are grouped together to form the context sets. They can be constructed thanks to similarity or proximity of objects (Thorisson, 1994), by the gestures that are made (Landragin, 2006) or by the discourse itself (Denis et al., 2006). We will be limited to the dimension of semantic *similarity* in this paper.

RDT claims that the context sets (*reference domains* or RD) are structures that both gather individuals and discriminate them. A reference domain is basically a set of objects that share some semantic description  $N$ . A partition that discriminates the elements is also attached to the domain. The partition is based on a *differentiation criterion* such that two elements being discriminated with this criterion are put in two different equivalency classes. For instance, in a domain of two buttons, one blue and one red, the two individuals share the same type and are differentiated with the color. While each one is “a button”, they can unambiguously be referred to with “the blue button” and “the red button” (or even shorter “blue”, “red”).

Different elements of a domain may be more or less focused/salient depending on the visual scene, or on the previous discriminations. We are assuming that the focus is defined as the most salient parts of the partition of a domain and can thus be represented as a *subset* of the partition. This is a binary state, that is, a part is focused or not. While it removes the possibility to have different degrees of focus inside a domain, it would help modeling a preference to focus similar objects together. We did not explore though the empirical relevance of this hypothesis.

We assume that each domain could be said more or less salient in a set of reference domains, called

a *referential space* or RS. The referential space is a storage for the domains that have been created so far. We consider here it is unique and shared. In the GIVE setting, the RS is actually not shared because the player does not know the maze *a priori* while the system knows it completely. But we assume that the RS is limited to the current room where the player is standing. Each time the player enters a new room, the RS is refreshed and a new one is built. We then suppose that the player is able to access the objects by walking around, and hence that the RS is shared, removing problems related to asymmetry.

The referential space provides a *traversal order* for the reference domains it contains. The most salient RD are tested first. While it would be interesting to model visual salience in the GIVE setting (Landragin, 2006), we are limited to equate salience and recency. Thus, each domain will be associated to a number indicating how recently it has been selected. The way the salience or the whole RS is affected by the discrimination process is described in section 3.2. We now provide a formal definition of a reference domain and a referential space building algorithm.

### 2.1 Reference domains

We assume that  $\langle E, V \rangle$  is an environment composed of  $E$ , the universe of all objects and  $V$ , the set of ground predicates that hold in the environment. *Props* is a set of unary predicates names such as blue, red, left, or right. *Types* is a set of types of unary predicates such as color, or position. We distinguish two disjoint subsets of *Types*, *Types<sub>pers</sub>* the *persistent* types, that are all the properties that describe permanently the objects, and *Types<sub>trans</sub>* the *transient* types, that are all the properties that change across time. *val* is the function  $val: Types \rightarrow 2^{Props}$  which maps a type on the predicates names, e.g.

$$val(color) = \{blue, red, green, yellow\}$$

A *reference domain*  $D$  is a tuple

$$\langle G_D, S_D, \sigma_D, (c, P, F) \rangle$$

where:

- $G_D \subseteq E$  is the set of objects of the domain, called the *ground of the domain*.
- $S_D \subseteq Props$  is the *semantic description* of the domain, such that  $\forall p \in S_D, \forall x \in G_D, p(x) \in V$ , that is,  $S_D$  is a description satisfied for all the elements of the ground.
- $\sigma_D \in \mathbb{N}$  is the *salience* of the domain

And  $(c, P, F)$  is a *partition structure* where:

- $c \in \text{Types}$  is a *differentiation criterion*
- $P$  is the *partition* generated by  $c$ , that is, if we define the equivalence relation

$$\mathcal{R}_c(x, y) \equiv \forall p \in \text{val}(c), p(x) \in V \Leftrightarrow p(y) \in V$$

then  $P = G_D / \mathcal{R}_c$ , i.e.  $P$  is the quotient set of  $G_D$  by  $\mathcal{R}_c$ .

- $F \subseteq P$  is the *focus* of  $P$ .

For instance, a domain composed of two buttons,  $b_1$  a blue button and  $b_2$  a red button, with a salience equal to 3, where  $b_1$  and  $b_2$  are differentiated using the color, and where  $b_1$  is in focus, would be noted as:

$$D = (\{\{b_1, b_2\}, \{button\}, 3, \\ (color, \{\{b_1\}, \{b_2\}\}, \{\{b_1\}\})\})$$

## 2.2 Referential space

The *referential space* RS is the set of existing domains. In the GIVE context, we assumed that it is both shared and refreshed each time the player enters a room. The initial construction of the RS consists in grouping all the objects of the room that are similar inside new reference domains. The RS can be viewed as a tree-like structure whose nodes are RD. The root node is a RD whose ground is all objects of the room. For a node domain  $D$ , and for each part of its partition which is not a singleton, there exists a child domain which discriminates the elements of the part. In other words, if a domain does not discriminate some individuals of its ground there exists another domain which does. Formally, the RS has to respect the following proposition where  $P_D$  denotes the partition of  $D$ .

$$\forall D \in RS, \forall P \in P_D, \\ |P| > 1 \Rightarrow \exists D' \in RS; G_{D'} = P \wedge |P_{D'}| > 1$$

In order to make sure that all the individuals could be discriminated, and thus focused, we introduce the default partition structure of a set  $X$ , which is a partition structure where the criterion is the identifier of objects and that contains then only singletons, we note  $\text{def}(X)$  the default partition of a set  $X$ , that is  $\text{def}(X) = (\text{id}, X / \mathcal{R}_{\text{id}}, \emptyset)$ .

To build initially the RS, the grouping algorithm (figure 1) is the following: it takes a list of types  $T$  ( $T_0$  means the head, and  $T_{1..n}$  the tail) which corresponds to different properties to group the objects. We are here only using the permanent properties of

objects, that is in GIVE their type and their color, ordered arbitrarily. It takes also an input domain which has a default partition. It then tries to partition the ground of this domain with the first property. If this property does not partition the ground, the next property is tested. If this property partitions the ground, a new domain is created for each non-singleton part of the partition, and the algorithm tries to partition it with the next property, so on recursively. We note  $\text{sh}(X, c)$  the set of properties of the type  $c$  that are shared by all elements of  $X$ :  $\text{sh}(X, c) = \{p | p \in \text{val}(c), \forall x \in X; p(x) \in V\}$ .

This partitioning algorithm is slightly different from the partitioning algorithm called  $\text{IA}_{part}$  found in (Gatt and van Deemter, 2007). First, it only partitions a set of objects using one unique property, whereas in  $\text{IA}_{part}$  the same set of objects can be partitioned several times. And second, while  $\text{IA}_{part}$  “destroys” the ground that is partitioned, our partitioning algorithm maintains both the ground and the partition attached to the domain.

```

1:  $RS \leftarrow RS \cup \{D\}$ 
2: if  $T \neq \emptyset$  then
3:    $P \leftarrow G_D / \mathcal{R}_{T_0}$ 
4:   if  $|P| = 1$  then
5:      $S_D \leftarrow S_D \cup \text{sh}(G_D, T_0)$ 
6:     createPartitions( $D, T_{1..n}, RS$ )
7:   else
8:     set  $(T_0, P, \emptyset)$  as  $D$ 's partition structure
9:     for all  $X \in P$  such that  $|X| > 1$  do
10:       $D' \leftarrow \langle X, S_D \cup \text{sh}(X, T_0), \sigma_D, \text{def}(X) \rangle$ 
11:      createPartitions( $D', T_{1..n}, RS$ )
12:     end for
13:   end if
14: end if

```

Figure 1: createPartitions( $D, T, RS$ )

## 3 Referring

In this section we detail the generation algorithm in RDT. It implements a dynamic view of referring whereby each referring act updates the current referential space. This incremental update of the referential space proceeds in three steps. First, a domain containing the referent is found. Then this domain is used to match a so called *underspecified domain* (Salmon-Alt and Romary, 2001). Third, the input RS is restructured relative to the selected reference domain.

The approach enables the implementation of a type B reversible reference module (Klarner, 2005), that is a module in which both directions share the

Expression	$U(N, t)$ matches $D$ iff $\exists(c, P, F) \in D$ ;
this one	$F = \{\{t\}\} \wedge \text{msd}(D)$
this N	$F = \{\{t\}\} \wedge t \in N^{\mathcal{I}}$
the N	$t \in N^{\mathcal{I}} \wedge \{t\} \in P \wedge \forall X \in P, X \neq \{t\} \Rightarrow X \cap N^{\mathcal{I}} = \emptyset$
the other one	$F \neq \emptyset \wedge P \setminus F = \{\{t\}\} \wedge \text{msd}(D)$
the other N	$F \neq \emptyset \wedge P \setminus F = \{\{t\}\} \wedge G_D \subseteq N^{\mathcal{I}}$
another one	$F \neq \emptyset \wedge \{t\} \in P \setminus F \wedge \text{msd}(D)$
another N	$F \neq \emptyset \wedge \{t\} \in P \setminus F \wedge G_D \subseteq N^{\mathcal{I}}$
a N	$t \in N^{\mathcal{I}} \wedge t \in G_D$

Table 1: Underspecified domains for each type of referring expression

same resources, namely a set of underspecified domains. In *interpretation*, the goal is to check for each existing domain if it matches the underspecified domain obtained from the referring expression. In *generation*, the idea is the opposite, that is, to check from an existing domain and a referent, which underspecified domain matches them.

We first introduce the different types of underspecified domains. We then present the overall referring algorithm and the process steering the continuous update of the referential space.

### 3.1 Underspecified domains

An underspecified domain (UD) represents a partially specified reference domain corresponding to the constraints carried by a referring act. We will say that an underspecified domain *matches* a reference domain if all the constraints of the UD are satisfied for the reference domain. There may be constraints on the ground of the domain, its salience or the existence of a particular partition structure. Table 1 summarizes most of the types of underspecified domains described in (Salmon-Alt and Romary, 2000; Salmon-Alt and Romary, 2001). Each underspecified domain is noted  $U(N, t)$ , where  $t$  is the intended referent and  $N \subseteq Props$  is a semantic description. We will note  $N^{\mathcal{I}}$  the set of objects that have the semantic description  $N$  that is  $N^{\mathcal{I}} = \{x | x \in E, \forall p \in N, p(x) \in V\}$ . We assume there is for each description  $N$  a given wording, and we will write for instance “the N” to denote a definite RE where N has to be replaced by the wording of  $N$ . The notation  $\text{msd}(D)$  stands for *most salient description*, that is, there is no more salient domain than  $D$  with a different description. This is equivalent to  $\nexists D' \in RS; \sigma_{D'} \geq \sigma_D \wedge S_{D'} \neq S_D$ .

The *indefinite* “a N” can always be generated but may be ambiguous. The only constraint placed on a domain by the corresponding UD is that it contains an element of type N. For example, the domain  $D_1 = \langle \{b_1, b_2, b_3\}, \{button\}, 0, (color, \{\{b_1, b_2\}, \{b_3\}\}, \emptyset) \rangle$

does not differentiate  $b_1$  from  $b_2$ , the only way we could access to  $b_1$  would be by uttering “a blue button”.

The *definite expression* “the N” requires that the target forms a semantically disjoint part in the reference domain partition. For example, in the above domain  $D_1$ , “the red button” can be used to refer to  $b_3$ .

Like the definite and indefinite, *the demonstrative* “this N” requires that the referent is of type N (belongs to  $N^{\mathcal{I}}$ ), but also requires the existence of a focused partition containing exactly the referent. For example, if a domain of blue buttons contains a partition structure such that  $P = \{\{b_1\}, \{b_2\}\}$ , it is possible to refer to  $b_1$  given that  $F = \{\{b_1\}\}$  by uttering “this blue button”, but it would not be the case if  $F = \{\{b_1\}, \{b_2\}\}$ .

*Alternative phrases such as “another/the other N”* both require that there is already something in focus which is not the referent. Definite alternative phrases require that the unfocused part of the partition contains exactly the target referent while indefinites only require that the unfocused part *contains* the referent. For example, if there is a domain of three blue buttons  $b_1, b_2, b_3$  with a partition structure such that  $F = \{\{b_2\}\}$ , it is possible to use the indefinite “another blue button” to refer to  $b_1$  while it would not be possible to use the definite “the other blue button”.

*One-anaphora of the form “this/another/the other one”* can be generated only if the description of the domain in which the referent has to be discriminated is already salient, in other words that  $\text{msd}(D)$  is true. For example, if the most salient domain in  $RS$  is a domain of blue buttons, it would not be possible to utter “this one” to refer to a red button inside a less salient domain.

### 3.2 Generation algorithm

The referring algorithm (figure 2) proceeds in three steps as follows.

The first step (line 1–2) determines in which reference domain, referring will be processed and thereby, which description will be used for instantiating the underspecified domains. The selected RD is the most salient RD with the smallest ground containing the target referent. If there are several such RD, an arbitrary one is picked. If the selected domain is  $D = \langle G_D, S_D, \sigma_D, (c, P, F) \rangle$ , then the description  $S$  used to instantiate the underspecified domain is the conjunction of the properties in the description  $S_D$  with the value of the differentiation criterion used to create the partition namely, properties of  $\text{val}(c)$  true of the referent (line 2). If the criterion is the identifier, it is ignored in  $S$ . For instance, if there is

a domain of buttons with a partition on color, the description might be  $\{button, blue\}$ .

In the second step, the algorithm iterates through the underspecified domains instantiated with  $S$  and selects the first that matches. The order in which underspecified domains are tested is particularly important. We use (Gundel et al., 1993) Givenness hierarchy and ordered the UD's based on the cognitive status of the corresponding referent. We extended the hierarchy to include alternative NPs: “this one” > “this N” > “the N” > “the other one” > “the other N” > “another one” > “another N” > “a N”.

In the third step, the referential space is restructured by either creating a new domain or increasing the salience of an existing domain (Figure 3). The goal of this restructuring step is to be able to restrict the further focus to a smaller domain. For instance, when dealing with red buttons we want to avoid focusing the blue buttons. The function first gathers all objects of  $D$  that have the persistent part of description  $S$  ( $G_p$  and  $S_p$ ), and if there is already a domain composed by these objects, its salience is increased such that it is the most salient (line 4). If there is no such domain, a new most salient domain is created with these objects and a default partition. Transient properties are not taken into account to regroup the objects because it would restrict too much further focus. For instance, limiting the restructuring to persistent properties avoids sequences like “Push the button on the right. Yeah this one”.

For example in a domain  $D$  containing a button  $b_1$  and a chair  $c_1$ ,

$$D = (\{b_1, c_1\}, \emptyset, 0, \\ \text{(objType, } \{\{b_1\}, \{c_1\}\}, \emptyset))$$

a reference to  $b_1$  could lead to the generation of the expression “the button”, the restructuring makes sure to create a new domain whose ground is only  $\{b_1\}$ . Therefore, we avoid producing unnecessary reference to the chair such as “Not this chair! Look for the button” (see section 4).

### 3.3 Dealing with plurals

The plurals treatment is quite similar to the singular cases, but we need to do two modifications to be able to generate plurals. The first modification is about the underspecified domains. Whereas we had individuals, here we want to generate an RE to a set of targets  $T = \{t_1..t_n\}$ . The UD's can easily be modified by just replacing every occurrence of  $\{t\}$  by  $T$  (and  $t \in N^I$  by  $T \subseteq N^I$ ). With this modification, we can only generate plurals for sets of

```

1:  $D \leftarrow$  most salient/specific domain containing  $t$ 
2:  $S \leftarrow S_D \cup \{p | p \in \text{val}(c), p(t) \in V\}$ 
3: for all  $U(S, t)$  sorted by Givenness do
4:   if  $U(S, t)$  matches  $D$  then
5:     restructure( $D, S, RS$ )
6:   return  $U(S, t)$ 
7:   end if
8: end for
9: return failure

```

Figure 2: generate( $t, RS$ )

```

1:  $S_p \leftarrow \{p | p \in S, \text{val}^{-1}(p) \in \text{Types}_{\text{pers}}\}$ 
2:  $G_p \leftarrow \{x | x \in G_D, \forall p \in S_p, p(x) \in V\}$ 
3: if  $\exists D' \in RS; G_{D'} = G_p$  then
4:    $\sigma_{D'} \leftarrow \max_{\sigma}(RS) + 1$ 
5: else
6:    $D' \leftarrow \langle G_p, S_p, \max_{\sigma}(RS) + 1, \text{def}(G_p) \rangle$ 
7:    $RS \leftarrow RS \cup \{D'\}$ 
8: end if

```

Figure 3: restructure( $D, S, RS$ )

objects that are parts of an existing partition. Imagine we have  $G_D = \{b_1, b_2, b_3, b_4\}$ , and a partition  $P = \{\{b_1, b_2\}, \{b_3, b_4\}\}$  then it is not possible to refer to  $\{b_2, b_3\}$  using a demonstrative because they cannot be focused together. It may be possible to adapt the UD to consider  $\bigcup F$  instead of  $F$ , that is for instance instead of  $F = \{T\}$  we would require that  $\bigcup F = T$ . But this possibility and its side-effects have not been yet explored.

The second modification is related to the generation algorithm and the description used to build the underspecified domains. Instead of retrieving the properties of the differentiation criterion for a single target we need to make sure that the properties are true for all the targets, that is (line 2), we need to have  $S \leftarrow S_D \cup \{p | p \in \text{val}(c), \forall t \in T, p(t) \in V\}$ .

## 4 Generation in the GIVE challenge

We present here how the generation module has been instantiated in the second edition of the GIVE challenge (Byron et al., 2007).

First, each time the player enters a new room, the partition algorithm is called on an initial domain  $D_r = \langle G_r, \emptyset, 0, \text{def}(G_r) \rangle$ , with  $G_r \subseteq E$  the set of all objects in the room, and the list of GIVE persistent types, that is *objType*, the type of objects, and *color*.

We then use the above referring algorithm in two ways. First, it is used to produce a first mention using only *persistent* properties and without updating the focus. Second, it is used to produce a series of

additional subsequent mentions whose function is to guide the player search. In this second step, *transient* spatial properties are used and the visual focus is continuously updated.

#### 4.1 First mention

The referring algorithm just described (cf. Figure 2) takes as input the current referential space  $RS$ , generates a referring expression for the target referent  $t$  and outputs a push instruction of the form “Push” $+v(\text{generate}(t, RS))$  where  $v$  is the verbalization function. Note that the referential space may contain domains with focused partitions coming from previous references to other objects, and therefore is not limited to producing definite or indefinite NPs.

#### 4.2 Subsequent mentions

All the subsequent mentions assume that the first mention has been performed but has not succeeded yet in identifying the referent. They are all based on focus and potentially on transient properties. The focus is defined as the set of visible objects. The algorithm (figure 4) first updates the focus of the partition of the most salient/specific domain  $D$  containing the target  $t$ . Then the rest of the algorithm generates different instructions depending on whether the target is or is not focused.

The lines 7–8 refine the focus using relative spatial properties of objects in their domain. It first computes these new properties  $hpos$  and  $vpos$  for all objects in  $\bigcup F$ , and adds them in  $V$ . The refinement is made by calling the partition function (algorithm 1) on a new domain  $D_F = \langle G_F, S_D, \sigma_D + 1, \text{def}(G_F) \rangle$ , using  $[hpos, vpos]$ . The salience of  $D_F$  is just higher than the salience of  $D$  such that  $D_F$  is preferred over  $D$  when generating. This refinement allows producing expressions like “the blue button on the right”. Because these properties are transient, they are erased from  $V$  after the generation and all the domains and partitions that may have been created using them including  $D_F$  are also erased.

Other lines produce expressions if the referent is not in focus. If there is nothing in focus, it produces “Look for X” where X is an RE for the referent. If there is something in focus which is not the referent, it first produces “Not X” where X is an RE designating what is in focus, then “Look for X” where X is an RE for the referent. Note that this is the only place where plurals can be generated (see section 3.3).

## 5 Detailed example

We present here a detailed example of the behavior of the reference module in the GIVE setting (Table 2). We assume that the player  $U$  enters a room with

```

1:  $D \leftarrow$  most salient/specific domain containing  $t$ 
2:  $F \leftarrow$  focus of the visible objects in  $D$ 
3:  $G_F \leftarrow \bigcup F$ 
4: if  $t \in G_F$  then
5:   if  $|G_F| > 1$  then
6:     computePositions( $G_F$ )
7:      $D_F \leftarrow \langle G_F, S_D, \sigma_D + 1, \text{def}(G_F) \rangle$ 
8:     createPartitions( $D_F$ ,  $[hpos, vpos]$ ,  $RS$ )
9:   end if
10:  return ‘Yeah!’ $+v(\text{generate}(t, RS))+$ ’!’
11: else
12:  if  $|G_F| = 0$  then
13:    return ‘Look for ’ $+v(\text{generate}(t, RS))$ 
14:  else
15:    return ‘Not ’ $+v(\text{generate}(G_F, RS))+$ ’!’
    Look for ’ $+v(\text{generate}(t, RS))+$ ’!’
16:  end if
17: end if

```

Figure 4: Algorithm to instruct the search for a referent

state of $U$	utterance of $S$
	Push a blue button ( $b_1$ )
see( $b_2$ )	Not this one! Look for the other one!
see( $b_1, b_2$ )	Yeah! The blue button on the right!
see( $b_1$ )	Yeah! This one!
push( $b_1$ )	
	Push the red button ( $b_3$ )
see( $b_3$ )	Yeah! This one!
push( $b_3$ )	
	Push the other blue button ( $b_2$ )

Table 2: Utterances produced by the system  $S$

three buttons, two blue buttons,  $b_1$  and  $b_2$  and a red button  $b_3$ .

### 5.1 Initializing the referential space

As soon as the player enters the room, the partition algorithm is called on the initial domain:

$$D_0 = \langle G_r, \emptyset, 0, \text{def}(G_r) \rangle$$

with  $G_r = \{b_1, b_2, b_3\}$ . The result is the  $RS$ :

$$\begin{aligned}
 D_0 = & \langle \{b_1, b_2, b_3\}, \{\text{button}\}, 0, \\
 & (\text{color}, \{\{b_1, b_2\}, \{b_3\}\}, \emptyset) \rangle \\
 D_1 = & \langle \{b_1, b_2\}, \{\text{button}, \text{blue}\}, 0, \\
 & (\text{id}, \{\{b_1\}, \{b_2\}\}, \emptyset) \rangle
 \end{aligned}$$

We will note the  $RS$  by grouping the domains that have the same salience and indicating the salience of a set of domains in subscript. That is, after the construction, the  $RS$  is:  $\{\{D_0, D_1\}_0\}$ .



## 5.2 “Push a blue button”

The system is first required to refer to  $b_1$ . As all the domains are equally salient, the algorithm tries to pick the most specific domain containing  $b_1$ , and it finds  $D_1$ . The description used to refer to  $b_1$  is the description of the domain  $S_{D_1} = \{button, blue\}$  and the value for the criterion which is the identifier and is then ignored. Inside  $D_1$  it then tries to refer to  $b_1$  by iterating through the underspecified domains to find the first one that matches  $D_1$ . Because there is no focus at this moment, the first found UD that matches is “a N”. It then performs restructuring of the  $RS$ , by trying to build a new subdomain of  $D_1$ . However, because there are only blue buttons in  $D_1$ , no subdomain is created and the salience of  $D_1$  is increased. Eventually, the expression is verbalized and “Push a blue button” is uttered. After this reference, the  $RS$  is then  $\{\{D_1\}_1, \{D_0\}_0\}$ .

## 5.3 “Not this one! Look for the other one!”

Before the subsequent mentions to  $b_1$  are made, the focus of the most salient/specific domain containing  $b_1$  is updated. We assume first that only  $b_2$  is visible, thus  $D_1$  becomes:

$$D_1 = \langle \{b_1, b_2\}, \{button, blue\}, 1, \\ (id, \{\{b_1\}, \{b_2\}\}, \{\{b_2\}\}) \rangle$$

According to the algorithm in figure 4, a reference to  $b_2$  has to be made first “Not  $b_2$ !”. Underspecified domains are iterated and the first that matches is “this one” considering that  $\{blue, button\}$  is the most salient description and  $b_2$  is in focus. No subdomain is created when restructuring the  $RS$ , only the salience of  $D_1$  is increased. The uttered expression is then “Not this one!”. As for the reference to  $b_1$ , the reference is still made in  $D_1$  and the first UD that matches is “the other one”. No restructuring apart from increasing salience is performed and the returned expression is eventually “Look for the other one!”. So, after referring to  $b_2$  and  $b_1$ , the  $RS$  is  $\{\{D_1\}_3, \{D_0\}_0\}$ .

## 5.4 “The blue button on the right”

We enjoined the player to turn around to search for  $b_1$ . We assume here that he did so and now can see both  $b_1$  and  $b_2$ . Before any reference can take place, the focus of  $D_1$  is updated:

$$D_1 = \langle \{b_1, b_2\}, \{button, blue\}, 3, \\ (id, \{\{b_1\}, \{b_2\}\}, \{\{b_1\}, \{b_2\}\}) \rangle$$

However, the focus can no more discriminate both buttons, and a refinement with the position has to be performed according to the algorithm 4. We assume that  $b_1$  is on the right while  $b_2$  is on the left. Positions are computed and new ground predicates are added to  $V$ :  $\{right(b_1), left(b_2)\}$ . A new domain  $D_2$  with a ground equal to the focus of  $D_1$ , that is  $\{b_1, b_2\}$ , is built and used as input for the partition algorithm. It is partitioned along the horizontal position ( $hpos$ ), and then added to the  $RS$ , that is:

$$D_2 = \langle \{b_1, b_2\}, \{button, blue\}, 4, \\ (hpos, \{\{b_1\}, \{b_2\}\}, \emptyset) \rangle$$

Before the reference to  $b_1$ , the  $RS$  is then  $\{\{D_2\}_4, \{D_1\}_3, \{D_0\}_0\}$ . A new reference to  $b_1$  is then made, but as  $D_2$  is more salient than  $D_1$  it is preferred for the reference. The first UD that matches is the definite “the N” built with the description  $\{button, blue, right\}$ , and “the blue button on the right” is uttered. However, because  $D_2$  was built with transient properties, it is erased from the  $RS$  and is recreated before each reference unless the player changes its visual focus.

## 5.5 “Yeah! This one!”

Now we assume that the player turned around again and only sees now  $b_1$ . The most salient/specific domain containing  $b_1$  is  $D_1$  and its focus is updated:

$$D_1 = \langle \{b_1, b_2\}, \{button, blue\}, 3, \\ (id, \{\{b_1\}, \{b_2\}\}, \{\{b_1\}\}) \rangle$$

The first matching UD is the demonstrative one-anaphora “this one”, no restructuring takes place except the increased salience of  $D_1$  and “Yeah! This one!” is produced. The  $RS$  is thus  $\{\{D_1\}_4, \{D_0\}_0\}$ .

## 5.6 “Push the red button”

We assume that given all these referring expressions, the player is at last able to push  $b_1$ . A new reference has to be made, this time to  $b_3$ , the red button. The most salient/specific domain containing  $b_3$  is actually  $D_0$ . In  $D_0$ , the first matching underspecified domain is the definite “the N”. The restructuring leads this time to create a new most salient domain  $D_3$  composed only of  $b_3$  (because it is the only red button):

$$D_3 = \langle \{b_3\}, \{button, red\}, 5, \\ (id, \{\{b_3\}\}, \emptyset) \rangle$$

The further reference to objects will thus avoid referring to something else than red buttons (see section 3.2). The  $RS$  is then  $\{\{D_3\}_5, \{D_1\}_4, \{D_0\}_0\}$ .

### 5.7 “Yeah! This one!”

Provided that  $D_3$  is now the most salient/specific container of  $b_3$ ,  $b_3$  can be focalized in the default partition of  $D_3$ , resulting in:

$$D_3 = (\{b_3\}, \{button, red\}, 5, (id, \{\{b_3\}\}, \{\{b_3\}\}))$$

The first matching UD is then “this one”, the restructuring just increases the salience of  $D_3$  and the system utters eventually “Yeah! This one!”. The  $RS$  is then  $\{\{D_3\}_6, \{D_1\}_4, \{D_0\}_0\}$ . Note that, even if the player would turn around and see  $b_1$  or  $b_2$  in the same time than  $b_3$ ,  $D_3$  being the current most salient/specific domain,  $b_1$  or  $b_2$  would not be focused.

### 5.8 “Push the other blue button”

We now have to refer to the last button  $b_2$ . The most salient/specific domain containing  $b_2$  is  $D_1$ , however  $D_1$  contains already a focus to  $b_1$ . Thus, the first matching UD is “the other N”. Note that we only considered visual focus, therefore the alternative anaphora “the other” does *not* refer to  $b_2$  because we already mentioned  $b_1$  but only because it is the last object the player saw in  $D_1$ . By chance, in the GIVE setting, the visual focus corresponds to the linguistic focus and thus uttering “Push the other blue button” sounds natural. It would be more complex to handle a setting with both the linguistic and the visual focus, but we think that the RDT is well-equipped to resolve this issue.

## 6 Evaluation

We evaluated the RDT generation model by comparing its performances with another system also competing in the GIVE challenge but based on a classical approach on (Dale and Haddock, 1991) that is restricted to generating definite and indefinite NPs. We designed a special evaluation world to test several reference cases, and for both approaches, we measured the average time from the moment of uttering a first mention designating a button to the moment of completion, that is when the button is successfully pushed. We also measured the average number of instructions that were provided in the meantime. The evaluation has been conducted with 30 subjects resulting in 20 valid games. The results show that the RDT performs better than the classical strategy, both for the average completion time (8.8 seconds versus 12.5 seconds) and for the number of instructions (6.4 versus 9.3). We conjecture that the good

results of RDT can be explained by the lower cognitive load resulting from the use of demonstrative NPs and one-anaphoras.

## 7 Other works and extensions

While some RE generation models focus on the side of generating the description itself (Dale and Reiter, 1995; Krahmer et al., 2003), we tried to focus more on the side of generating the determiner. While works such as (Poesio et al., 1999) also generates the determiner, they rely on statistical learning of this determiner. On the contrary we did so by representing logically the constraints carried by a referring expressions on the context of its interpretation. However, the presented model has several limits. First, as (Landragin and Romary, 2003) describe, there is no one-to-one relation between the referring expressions and the referring modes. In order to tackle this problem we can associate a *set* of UD to a referring expression. We only need then to add an additional loop on the different UD for a given type of referring expression. The second extension is the possibility to have several partitions. It is also possible to iterate over the set of partitions of a domain, but we then need to consider the salience of each partition. In addition, the restructuring has to be amended to increase the salience of the partition in which a generation is made.

## 8 Conclusions

We presented a reference generation algorithm based on Reference Domain Theory. The main improvement of this algorithm over existing approaches is the construction and update of a set of local contexts called a referential space. Each local context (reference domain) can be used as a context for referring. The dynamic aspect of the reference process consists both in the continuous update of the reference domains and in the update of the referential space. Thus, the presented algorithm can generate a variety of referring expressions ranging from definite, indefinite to demonstrative, alternative phrases, one-anaphora and plurals. The instantiation in the GIVE challenge was a baptism for the generation algorithm and the GIVE setting offered us a good opportunity to test the serial nature of the reference process. It enabled us to evaluate the RDT approach and proved that it is successful.

*We would like to thank Luciana Benotti, Claire Gardent, and the people participating to the GIVE challenge at the LORIA for their help during the model development. We also would like to thank the anonymous reviewers for their precious insights.*

## References

- Donna K. Byron, Alexander Koller, Jon Oberlander, Laura Stoia, and Kristina Striegnitz. 2007. Generating instructions in virtual environments (GIVE): A challenge and an evaluation testbed for NLG. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, Washington, DC.
- Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 165–173, Athens, Greece, March. Association for Computational Linguistics.
- Robert Dale and Nicholas J. Haddock. 1991. Generating referring expressions involving relations. In *Proceedings of the 5th Conference of the European Chapter of the ACL, EACL-91*.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Alexandre Denis, Guillaume Pitel, and Matthieu Quignard. 2006. Resolution of Referents Groupings in Practical Dialogues. In *Proceedings of the 7th SIGDial Workshop on Discourse and Dialogue - SIGdial'06*, Sydney Australia.
- Albert Gatt and Kees van Deemter. 2007. Incremental generation of plural descriptions: Similarity and partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP-07*.
- Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307.
- Martin Klarner. 2005. Reversibility and re-usability of resources in NLG and natural language dialog systems. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, Aberdeen, Scotland.
- Emiel Kraemer and Marit Theune. 2002. Efficient context-sensitive generation of referring expressions. In K. van Deemter and R. Kibble, editors, *Information sharing: Givenness and newness in language processing*, pages 223–264. CSLI Publications, Stanford.
- Emiel Kraemer, Sebastiaan van Erk, and Andr Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 23:53–72.
- Frédéric Landragin and Laurent Romary. 2003. Referring to Objects Through Sub-Contexts in Multimodal Human-Computer Interaction. In *Proceedings of the Seventh Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck'03)*, pages 67–74. Saarland University.
- Frédéric Landragin. 2006. Visual perception, language and gesture: A model for their understanding in multimodal dialogue systems. *Signal Processing*, 86(12):3578–3595.
- Massimo Poesio, Renate Henschel, Janet Hitzeman, and Rodger Kibble. 1999. Statistical NP generation: A first report. In *Proceedings of the ESSLLI Workshop on NP Generation*, Utrecht.
- Anne Reboul. 1998. A relevance theoretic approach to reference. In *Acts of the Relevance Theory Workshop*, University of Luton, England.
- Susanne Salmon-Alt and Laurent Romary. 2000. Generating referring expressions in multimodal contexts. In *Workshop on Coherence in Generated Multimedia - INLG 2000*, Mitzpe Ramon, Israel.
- Susanne Salmon-Alt and Laurent Romary. 2001. Reference resolution within the framework of cognitive grammar. In *Proceeding of the International Colloquium on Cognitive Science*, San Sebastian, Spain.
- Robert Stalnaker. 1998. On the representation of context. *Journal of Logic, Language and Information*, 7(1):3–19.
- Kristinn R. Thorisson. 1994. Simulated perceptual grouping: An application to human-computer interaction. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Atlanta, Georgia.



# Hierarchical Reinforcement Learning for Adaptive Text Generation

**Nina Dethlefs**

University of Bremen, Germany  
dethlefs@uni-bremen.de

**Heriberto Cuayáhuil**

University of Bremen, Germany  
heriberto@uni-bremen.de

## Abstract

We present a novel approach to natural language generation (NLG) that applies hierarchical reinforcement learning to text generation in the wayfinding domain. Our approach aims to optimise the integration of NLG tasks that are inherently different in nature, such as decisions of content selection, text structure, user modelling, referring expression generation (REG), and surface realisation. It also aims to capture existing interdependencies between these areas. We apply hierarchical reinforcement learning to learn a generation policy that captures these interdependencies, and that can be transferred to other NLG tasks. Our experimental results—in a simulated environment—show that the learnt wayfinding policy outperforms a baseline policy that takes reasonable actions but without optimization.

## 1 Introduction

Automatic text generation involves a number of sub-tasks. (Reiter and Dale, 1997) list the following as core tasks of a complete NLG system: content selection, discourse planning, sentence planning, sentence aggregation, lexicalisation, referring expression generation and linguistic realisation. However, decisions made for each of these core tasks are not independent of each other. The value of one generation task can change the conditions of others, as evidenced by studies in corpus linguistics, and it can therefore be undesirable to treat them all as isolated modules. In this paper, we focus on inter-related decision making in the areas of content selection, choice of text structure, referring expression

and surface form. Concretely, we generate route instructions that are tailored specifically towards different user types as well as different environmental features. In addition, we aim to balance the degree of variation and alignment in texts and produce lexical and syntactic patterns of co-occurrence that resemble those of human texts of the same domain. Evidence for the importance of this is provided by (Halliday and Hasan, 1976) who note the way that lexical cohesive ties contribute to text coherence as well as by the theory of interactive alignment. According to (Pickering and Garrod, 2004) we would expect significant traces of lexical and syntactic self-alignment in texts.

Approaches to NLG in the past have been either rule-based (Reiter and Dale, 1997) or statistical (Langkilde and Knight, 1998). However, the former relies on a large number of hand-crafted rules, which makes it infeasible for controlling a large number of interrelated variables. The latter typically requires training on a large corpus of the domain. While these approaches may be better suitable for larger domains, for limited domains such as our own, we propose to overcome these drawbacks by applying Reinforcement Learning (RL)—with a hierarchical approach. Previous work that has used RL for NLG includes (Janarthanam and Lemon, 2009) who employed it for alignment of referring expressions based on user models. Also, (Lemon, 2008; Rieser and Lemon, 2009) used RL for optimising information presentation styles for search results. While both approaches displayed significant effects of adaptation, they focused on a single area of optimisation. For larger problems, however, such as the one we are aiming to solve, flat RL will not be applicable due to the large state space. We therefore sug-

gest to divide the problem into a number of subproblems and apply hierarchical reinforcement learning (HRL) (Barto and Mahadevan, 2003) to solve it.

We describe our problem in more detail in Section 2, our proposed HRL architecture in Sections 3 and 4 and present some results in Section 5. We show that our learnt policies outperform a baseline that does not adapt to contextual features.

## 2 Generation tasks

Our experiments are all drawn from an indoor navigation dialogue system which provides users with route instructions in a university building and is described in (Cuayáhuitl et al., 2010). We aim to optimise generation within the areas of *content selection*, *text structure*, *referring expression generation* and *surface realisation*.

**Content Selection** Content selection decisions are subject to different user models. We distinguish users who are familiar with the navigation environment and users who are not. In this way, we can provide different routes for these users corresponding to their particular information need. Specifically, we provide more detail for unfamiliar than familiar users by adding any or several of the following: (a) landmarks at decision points, (b) landmarks lying on long route segments, (c) specifications of distance.

**Text Structure** Depending on the type of user and the length of the route, we choose among three different text generation strategies to ease the cognitive load of the user. Examples of all strategies are displayed in Table 1. All three types resulted from an analysis of a corpus of 24 human-written driving route instructions. We consider the first type (sequential) most appropriate for long or medium-long routes and both types of user. The second type (temporal) is appropriate for unfamiliar users and routes of short or medium length. It divides the route into an explicit sequence of consecutive actions. The third type (schematic) is used in the remaining cases.

**Referring Expression Generation** We distinguish three types of referring expressions: *common names*, *familiar names* and *descriptions*.

In this way, entities can be named according to the users’ prior knowledge. For example, one and the same room can be called either ‘the student union room’, ‘room A3530’ or ‘the room right at the corner beside the entrance to the terrace’.

**Surface Realisation** For surface realisation, we aim to generate texts that display a natural balance of (self-)alignment and variation. While it is a rule of writing that texts should typically contain variation of surface forms in order not to appear repetitive and stylistically poor, there is evidence that humans also get influenced by self-alignment processes during language production. Specifically, (Garrod and Anderson, 1987; Pickering and Garrod, 2004) argue that the same mental representations are used during language production and comprehension, so that alignment occurs regardless of whether the last utterance was made by another person or by the speaker him- or herself (for experimental evidence see (Branigan et al., 2000; Bock, 1986)). We can therefore hypothesise that coherent texts will, besides variation, also display a certain degree of self-alignment. In order to determine a proper balance of alignment and variation, we computed the degree of lexical repetition from our corpus of 24 human route descriptions. This analysis was based on (Hirst and St-Onge, 1998) who retrieve lexical chains from texts by identifying a number of relations between lexical items. We focus here exclusively on Hirst & St-Onge’s ‘extra-strong’ relations, since these can be computed from shallow properties of texts and do not require a large corpus of the target domain. In order to make a fair comparison between the human texts and our own, we used a part-of-speech (POS) tagger (Toutanova and Manning, 2000)<sup>1</sup> to extract those grammatical categories that we aim to control within our framework, i.e. nouns, verbs, prepositions, adjectives and adverbs. Based on these categories, we compute the proportion of tokens that are members in lexical chains, the ‘alignment score’ (AS), according to the following equation:

$$AS = \frac{\text{Lexical tokens in chains}}{\text{Total number of tokens}} \times 100. \quad (1)$$

We obtained an average alignment score of 43.3% for 24 human route instructions. In contrast, the

<sup>1</sup><http://nlp.stanford.edu/software/tagger.shtml>

Table 1: *Different text generation strategies for the same underlying route.*

Type 1: Sequential	Type 2: Temporal	Type 3: Schematic
Turn around, and go straight to the glass door in front of you. Turn right, then follow the corridor until the lift. It will be on your left-hand side.	First, turn around. Second, go straight to the glass door in front of you. Third, turn right. Fourth, follow the corridor until the lift. It will be on your left-hand side.	<ul style="list-style-type: none"> <li>- Turn around.</li> <li>- Go straight until the glass door in front of you. (20 m)</li> <li>- Turn right</li> <li>- Follow the corridor until the lift. (20 m)</li> <li>- It will be on your left-hand side.</li> </ul>

same number of instructions generated by Google Maps yielded 78.7%, i.e. an almost double amount of repetition. We will therefore train our agent to generate texts with an about medium alignment score.

### 3 Hierarchical Reinforcement Learning for NLG

The idea of *text generation as an optimization problem* is as follows: given a set of generation states, a set of actions, and an objective reward function, an optimal generation strategy maximizes the objective function by choosing the actions leading to the highest reward for every reached state. Such states describe the system’s knowledge about the generation task (e.g. content selection, text structure, REG, surface realization). The action set describes the system’s capabilities (e.g. `expand_sequential_aggregation`, `expand_schematic_aggregation`, `expand_lexical_items`, etc.). The reward function assigns a numeric value for each taken action. In this way, text generation can be seen as a finite sequence of states, actions and rewards  $\{s_0, a_0, r_1, s_1, a_1, \dots, r_{t-1}, s_t\}$ , where the goal is to find an optimal strategy automatically. To do that we use hierarchical reinforcement learning in order to optimize a hierarchy of text generation policies rather than a single policy.

The hierarchy of RL agents consists of  $L$  levels and  $N$  models per level, denoted as  $M = M_j^i$ , where  $j \in \{0, \dots, N - 1\}$  and  $i \in \{0, \dots, L - 1\}$ . Each agent of the hierarchy is defined as a Semi-Markov Decision Process (SMDP) consisting of a 4-tuple  $\langle S_j^i, A_j^i, T_j^i, R_j^i \rangle$ .  $S_j^i$  is a set of states,  $A_j^i$  is a set of actions, and  $T_j^i$  is a transition function that determines the next state  $s'$  from the current state  $s$  and the performed action  $a$  with a probability of

$P(s'|s, a)$ .  $R_j^i(s', \tau|s, a)$  is a reward function that specifies the reward that an agent receives for taking an action  $a$  in state  $s$  at time  $\tau$ . Since SMDPs allow for temporal abstraction, that is, actions may take a variable number of time steps to complete, the random variable  $\tau$  represents this number of time steps. Actions can be either primitive or composite. The former yield single rewards, the latter (executed using a stack mechanism) correspond to SMDPs and yield cumulative discounted rewards. The goal of each SMDP is to find an optional policy  $\pi^*$  that maximises the reward for each visited state, according to

$$\pi_j^{*i}(s) = \arg \max_{a \in A} Q_j^{*i}(s, a). \quad (2)$$

where  $Q_j^i(s, a)$  specifies the expected cumulative reward for executing action  $a$  in state  $s$  and then following  $\pi^*$ . For learning a generation policy, we use hierarchical Q-Learning (HSMQ) (Dietterich, 1999). The dynamics of SMDPs are as follows: when an SMDP terminates its execution, it is popped off the stack of models to execute, and control is transferred to the next available SMDP in the stack, and so on until popping off the root SMDP. An SMDP terminates when it reaches one of its terminal states. This algorithm is executed until the Q-values of the root agent stabilize. The hierarchical decomposition allows to find context-independent policies with the advantages of policy reuse and facilitation for state-action abstraction. This hierarchical approach has been applied successfully to dialogue strategy learning (Cuayahuitl et al., 2010).

## 4 Experimental Setting

### 4.1 Hierarchy of SMDPs

The hierarchy consists of 15 agents. It is depicted in Figure 1. The root agent is responsible for deter-

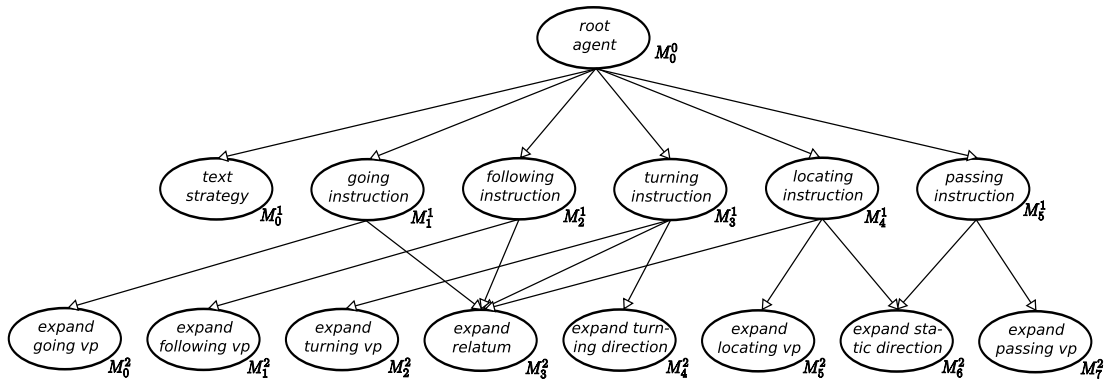


Figure 1: Hierarchy of agents for learning adaptive text generation strategies in the wayfinding domain

mining a route instruction type for a navigation situation. We distinguish turning, passing, locating, going and following instructions. It also chooses a text generation strategy and the information structure of the clause (i.e., marked or unmarked theme (Halliday and Matthiessen, 2004)). Leaf agents are responsible for expanding constituents in which variation or alignment can occur, e.g. the choice of verb or prepositional phrase.

#### 4.2 State and action sets

We distinguish three kinds of state representations, displayed in Table 2. The first ( $M_{10}^0$  and  $M_0^1$ ) encodes information on the spatial environment and user type so that texts can be tailored towards these variables. These variables play a major part in our simulated environment (Section 5.1). The second representation ( $M_1^1 - M_5^1$  and  $M_3^2$ ) controls sentence structure and ensures that all required constituents for a message have been realised. The third (all remaining models) encodes variants of linguistic surface structure and represents the degree of alignment of all variants. We address the way that these alignment values are computed in Section 4.4. Actions can be either primitive or composite. Whereas the former expand a logical form directly, the latter correspond to SMDPs at different levels of the hierarchy. All parent agents have both types of actions, only the leaf agents have exclusively primitive actions. The set of primitive actions is displayed in Table 2, all composite actions, corresponding to models, are shown in Figure 1. The average number of state-action pairs for a model is  $|S \times A| = 77786$ . While in the present work, the action set was de-

termined manually, future work can aim at learning hierarchies of SMDPs automatically from data.

#### 4.3 Prior Knowledge

Agents contain prior knowledge of two sorts. First, the root agents and agents at the first level of the hierarchy contain prior probabilities of executing certain actions. For example, given an unfamiliar user and a long route, model  $M_0^1$ , *text\_strategy*, is initiated with a higher probability of choosing a sequential text strategy than a schematic or temporal strategy. Second, leaf agents of the hierarchy are initiated with values of a hand-crafted language model. These values indicate the probabilities of occurrence of the different surface forms of the leaf agents listed in Table 2. Both types of prior probabilities are used by the reward functions described below.

#### 4.4 Reward functions

We use two types of reward function, both of which are directly motivated by the principles we stated in Section 2. The first addresses interaction length (the shorter the better) and the choice of actions tailored towards the user model and spatial environment.

$$R = \begin{cases} 0 & \text{for reaching the goal state} \\ -10 & \text{for an already invoked subtask} \\ p(a) & \text{otherwise} \end{cases} \quad (3)$$

$p(a)$  corresponds to the probability of the last action given the current state, described above as prior knowledge. The second reward function addresses



Table 2: State and action sets for learning adaptive text generation strategies in the wayfinding domain

Model	State Variables	Action Set
$M_0^0$	text_strategy (FV), info_structure (FV), instruction (FV), slot_in_focus(0=action, 1=landmark), user_type(0=unfamiliar, 1=familiar) subtask_termination(0=continue, 1=halt)	expand_text_strategy ( $M_0^1$ ), turning ( $M_3^2$ ), going ( $M_1^1$ ), passing ( $M_5^2$ ), following ( $M_2^2$ ), locating_instr. ( $M_4^2$ ), expand_unmarked_theme
$M_0^1$	end(0=continue, 1=halt), text_strategy (FV), route_length (0=short, 1=medium, 2=long), user_type(0=unfam., 1=fam.)	expand_schematic_aggregation, expand_sequence_aggregation, expand_temporal_aggregation
$M_1^1$	going_vp (FV), limit (FV), SV	expand_going_vp ( $M_0^2$ ), expand_limit
$M_2^1$	following_vp (FV), SV, limit (FV)	expand_following_vp ( $M_1^2$ ), expand_limit
$M_3^1$	turning_location (FV), turning_vp (FV), SV, turning_direction (FV)	expand_turning_vp ( $M_2^3$ ), expand_turning_loc., expand_turning_direction ( $M_4^3$ )
$M_4^1$	np_locatum (FV), locating_vp (FV), static_direction (FV), SV	expand_np_locatum, expand_locating_vp ( $M_5^2$ ), expand_static_dir. ( $M_6^2$ )
$M_5^1$	np_locatum (FV), passing_vp (FV), SV, static_direction (FV)	expand_pass._vp ( $M_6^2$ ), expand_static_dir. ( $M_6^2$ )
$M_0^2$	vp_go_straight_ahead, vp_go_straight, vp_move_straight_ahead, vp_walk_straight_ahead, vp_walk_straight (all AS)	Actions correspond to expansions of lexemes
$M_1^2$	vp_follow, vp_go_over, vp_walk_down, vp_go_down, vp_go_up, vp_walk_up, vp_walk_over (all AS)	Actions correspond to expansions of lexemes
$M_2^2$	vp_walk, vp_veer, vp_hang, vp_bear (all AS), vp_go, vp_head, vp_turn (all AS)	Actions correspond to expansions of lexemes
$M_3^2$	identifiability(0=not id., 1=id.), user_type(0=un-, fam., 1=fam.), relatum_identifiability (FV), relatum_name (FV)	expand_relatum_id., expand_relatum, _not_id., expand_descriptive, expand_common_name
$M_4^2$	pp_nonphoric, pp_nonphoric_handedness, pp_nonphoric_poss, pp_phoric pp_nonphoric_side (all AS)	Actions correspond to expansions of lexemes
$M_5^2$	vp_be, vp_be_located_at, vp_get_to, vp_see (all AS)	Actions correspond to expansions of lexemes
$M_6^2$	direction_on, direction_poss, direction_to (all AS)	Actions correspond to expansions of lexemes
$M_7^2$	vp_move_past, vp_pass, vp_pass_by, vp_walk_past (all AS)	Actions correspond to expansions of lexemes

(FV = filling status): 0=unfilled, 1=filled. (SV = shared variables): the variables np\_actor (FV), relatum (FV), sentence (FV) and information\_need (0=low, 1=high) are shared by several subagents; the same applies to their corresponding expansion actions. (AS = alignment score): 0=unaligned, 1=low AS, 2=medium AS, 3=high AS.

the tradeoff between alignment and variation:

$$R = \begin{cases} 0 & \text{for reaching the goal state} \\ p(a) & \text{for medium alignment} \\ -0.1 & \text{otherwise} \end{cases} \quad (4)$$

Whilst the former reward function is used by the root and models  $M_0^1$  -  $M_5^1$  and  $M_2^2$ , the latter is used by models  $M_0^2$  -  $M_1^2$  and  $M_3^2$  -  $M_7^2$ . It rewards the agent for a medium alignment score, which corresponds to the score of typical human texts we computed in Section 2. The alignment status of a constituent is computed by the Constituent Alignment Score (CAS) as follows, where MA stands for ‘medium

alignment’.

$$CAS(a) = \frac{\text{Count of occurrences}(a)}{\text{Occurrences of } a \text{ without MA}} \quad (5)$$

From this score, we can determine the degree of alignment of a constituent by assigning ‘no alignment’ for a constituent with a score of less than 0.25, ‘low alignment’ for a score between 0.25 and 0.5, ‘medium alignment’ for a score between 0.5 and 0.75 and ‘high alignment’ above. On the whole thus, the agent’s task consists of finding a balance between choosing the most probable action given the language model and choosing an action that aligns with previous utterances.

## 5 Experiments and Results

### 5.1 Simulated Environment

The simulated environment encodes information on the current user type (un-/familiar with the environment) and corresponding information need (low or high), the length of the current route (short, medium-long, long), the next action to perform (turn, go straight, follow a path, pass a landmark or take note of a salient landmark) and the current focus of attention (the action to be performed or some salient landmark nearby). Thus, there are five different state variables with altogether 120 combinations, sampled from a uniform distribution. This simple form of stochastic behaviour is used in our simulated environment. Future work can consider inducing a learning environment from data.

### 5.2 Comparison of learnt and baseline policies

In order to test our framework, we designed a simulated environment that simulates different navigational situations, routes of different lengths and different user types. We trained our HRL agent for 10,000 episodes with the following learning parameters: the step-size parameter  $\alpha$  was initiated with 1 and then reduced over time by  $\alpha = \frac{1}{1+t}$ ,  $t$  being the time step. The discount rate parameter  $\gamma$  was 0.99 and the probability of random action  $\epsilon$  was 0.01 (see (Sutton and Barto, 1998) for details on these parameters). Figure 2 compares the learnt behaviour of our agent with a baseline (averaged over 10 runs) that chooses actions at random in models  $M_0^1$  and  $M_0^2 - M_7^2$  (i.e., the baseline does not adapt its text strategy to user type or route length and neither performs adaptation of referring expressions or alignment score). The user study reported in (Cuayáhuil et al., 2010) provided users with instruction using this baseline generation behaviour. The fact that users had a user satisfaction score of 90% indicates that this is a sensible baseline, producing intelligible instructions. We can observe that after a certain number of episodes, the performance of the trained agent begins to stabilise and it consistently outperforms the baseline.

## 6 Example of generation

As an example, Figure 3 shows in detail the generation steps involved in producing the clause ‘Follow

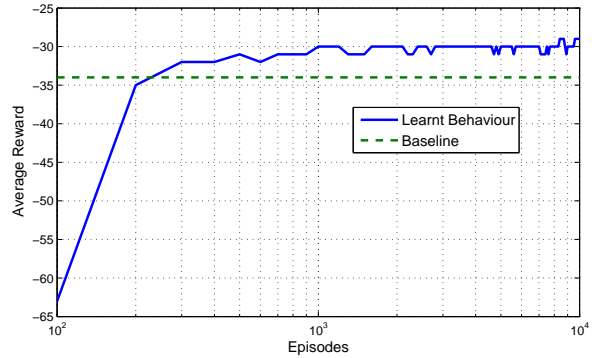


Figure 2: Comparison of learnt and baseline behaviour in the generation of route descriptions

the corridor until the copyroom’ for an unfamiliar user and a route of medium length. Generation starts with the root agent in state (0,0,0,0,0,0), which indicates that text\_strategy, info\_structure and instruction are unfilled slots, the slot\_in\_focus of the sentence is an action, the status of subtask\_termination is ‘continue’ and the user\_type is unfamiliar. After the primitive action expand\_unmarked\_theme was executed, the state is updated to (0,1,0,0,0,0), indicating the filled slot. Next, the composite action text\_strategy is executed, corresponding to model  $M_0^1$ . The initial state (1,0,0) indicates a route of medium length, an unfilled text\_strategy slot and an unfamiliar user. After the primitive action expand\_sequential\_text was chosen, the terminal state is reached and control is returned to the root agent. Here, the next action is following\_instruction corresponding to model  $M_2^1$ . The initial state (0,1,0,0,0,0) here indicates unfilled slots for following\_vp, np\_actor, sentence, path, limit and relatum, as well as a high information\_need of the current user. The required constituents are expanded in turn. First, the primitive actions expand\_limit, expand\_np\_actor, expand\_s and expand\_path cause their respective slots in the state representation to be filled. Next, the composite action expand\_relatum is executed with an initial state (0,1,0,0) representing an identifiable landmark, unfilled slots for a determiner and a referring expression for the landmark and an unfamiliar user. Two primitive actions, expand\_relatum\_identifiable and expand\_relatum\_common\_name, cause the agent to

reach its terminal state. The generated referring expression thus treats the referenced entity as either known or easily recoverable. Finally, model  $M_1^2$  executes the composite action `expand_following_vp`, which is initialised with a number of variables corresponding to the alignment status of different verb forms. Since this is the first time this agent is called, none of them shows traces of alignment (i.e., all values are 0). Execution of the primitive action `expand_following_vp` causes the respective slot to be updated and the agent to terminate. After this sub-task, model  $M_2^1$  has also reached its terminal state and control is returned to the root agent.

As a final step towards surface generation, all chosen actions are transformed into an SPL (Kasper, 1989). The type ‘following instruction’ leads to the initialisation of a semantically underspecified scaffold of an SPL, all other actions serve to supplement this scaffold to preselect specific syntactic structures or lexical items. For example, the choice of ‘`expand_following_vp`’ leads to the lexical item ‘follow’ being inserted. Similarly, the choice of ‘`expand_path`’ leads to the insertion of ‘the corridor’ into the SPL to indicate the path the user should follow. ‘`expand_limit`’, in combination with the choice of referring expression, leads to the insertion of the PP ‘until the copy room’. For generation of more than one instruction, aggregation has to take place. This is done by iterating over all instructions of a text and inserting them into a larger SPL that realises the aggregation. Finally, the constructed SPL is passed to the KPML surface generator (Bateman, 1997) for string realisation.

## 7 Discussion

We have argued in this paper that HRL is an especially suited framework for generating texts that are adaptive to different users, to environmental features and properties of surface realisation such as alignment and variation. While the former tasks appear intuitively likely to contribute to users’ comprehension of texts, it is often not recognised that the latter task can have the same effect. Differing surface forms of identical concepts in texts without motivation can lead to user confusion and deteriorate task success. This is supported by Clark’s ‘principle of contrast’ (Clark, 1987), according to which

new expressions are only introduced into an interaction when the speaker wishes to contrast them with other entities already present in the discourse. Similarly, a study by (Clark and Wilkes-Gibbs, 1986) showed that interlocutors tend to align their referring expressions and thereby achieve more efficient and successful dialogues. We tackled the integration of different NLG tasks by applying HRL and presented results, which showed to be promising. As an alternative to RL, other machine learning approaches may be conceivable. However, supervised learning requires a large amount of training data, which may not always be available, and may also produce unpredictable behaviour in cases where a user deviates from the behaviour covered by the corpus (Levin et al., 2000). Both arguments are directly transferable to NLG. If an agent is able to act only on grounds of what it has observed in a training corpus, it will not be able to react flexibly to new state representations. Moreover, it has been argued that a corpus for NLG cannot be regarded as an equivalent gold standard to the ones of other domains of NLP (Belz and Reiter, 2006; Scott and Moore, 2006; Viethen and Dale, 2006). The fact that an expression for a semantic concept does not appear in a corpus does not mean that it is an unsuited or impossible expression. Another alternative to pure RL is to apply semi-learned behaviour, which can be helpful for tasks with very large state-action spaces. In this way, the state-action space is reduced to only sensible state-action pairs by providing the agent with prior knowledge of the domain. All remaining behaviour continues to be learnt. (Cuayáhuil, 2009) suggests such an approach for learning dialogue strategies, but again the principle is transferable to NLG. While there is room for exploration of different RL methods, it is clear that neither traditional rule-based accounts of generation, nor  $n$ -gram-based generators can achieve the same flexible generation behaviour given a large, and partially unknown, number of state variables. Since state spaces are typically very large, specifying rules for each single condition is at best impractical. Especially for tasks such as achieving a balanced alignment score, as we have shown in this paper, decisions depend on very fine-grained textual cues such as patterns of co-occurrence which are hard to pin down accurately by hand. On the other hand, statistical approaches

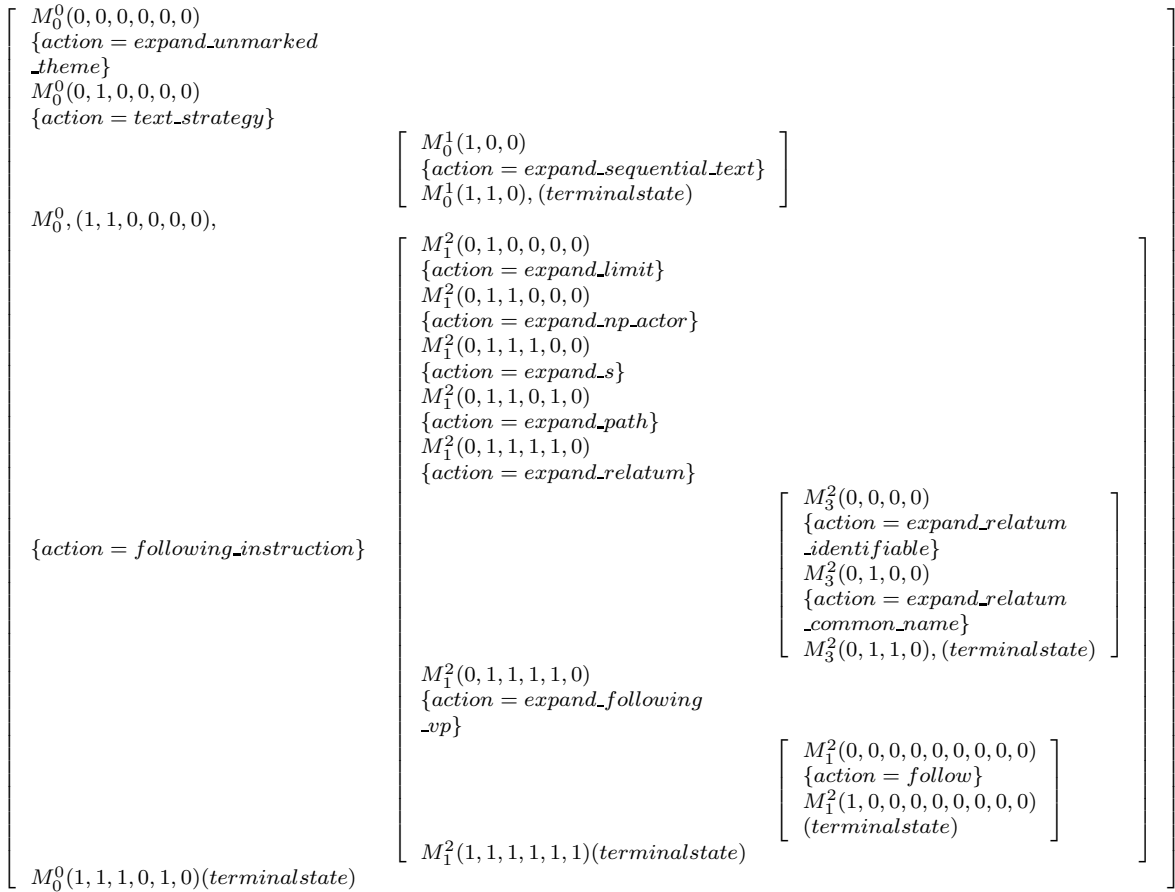


Figure 3: Example of generation for the clause ‘Follow the corridor until the copy room’. This example shows decision making for a single instruction, adaptation and alignment occurs over longer sequences of text.

to generation that are based on  $n$ -grams focus on the frequency of constructions in a corpus without taking contextual variables such as user type or environmental properties into account. Further, they share the problem of supervised learning approaches discussed above, namely, that it can act only on grounds of what it has observed in the past, and are not well able to adapt to novel situations. For a more detailed account of statistical and trainable approaches to NLG as well as their advantages and drawbacks, see (Lemon, 2008).

## 8 Conclusion

We presented a novel approach to text generation that applies hierarchical reinforcement learning to optimise the following interrelated NLG tasks: content selection, choice of text structure, referring ex-

pressions and surface structure. Generation decisions in these areas were learnt based on three different variables: the type of user, the properties of the spatial environment and the proportion of alignment and variation in texts. Based on a simulated environment, we compared the results of different policies and demonstrated that the learnt policy outperforms a baseline that chooses actions without taking contextual variables into account. Future work can transfer our approach to different domains of application or to other NLG tasks. In addition, our preliminary simulation results should be confirmed in an evaluation study with real users.

## Acknowledgements

This work was partly supported by DFG SFB/TR8 “Spatial Cognition”.

## References

- Barto, A. G. and Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning. *Discrete Event Dynamic Systems*, 13:2003.
- Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55.
- Belz, A. and Reiter, E. (2006). Comparing automatic and human evaluation of nlg systems. In *In Proc. EACL06*, pages 313–320.
- Bock, K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, 18.
- Branigan, H. P., Pickering, M. J., and Cleland, A. (2000). Syntactic coordination in dialogue. *Cognition*, 75.
- Clark, E. (1987). The principle of contrast: A constraint on language acquisition. In MacWhinney, B., editor, *Mechanisms of Language Acquisition*, pages 1–33. Lawrence Erlbaum Assoc., Hillsdale, NJ.
- Clark, H. H. and Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, 22.
- Cuayáhuitl, H. (2009). *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. PhD thesis, School of Informatics, University of Edinburgh.
- Cuayáhuitl, H., Dethlefs, N., Richter, K.-F., Tenbrink, T., and Bateman, J. (2010). A dialogue system for indoor wayfinding using text-based natural language. *International Journal of Computational Linguistics and Applications*, ISSN 0976-0962.
- Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2010). Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech and Language*, 24(2):395–429.
- Dietterich, T. G. (1999). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303.
- Garrod, S. and Anderson, A. (1987). Saying What You Mean in Dialogue: A Study in conceptual and semantic co-ordination. *Cognition*, 27.
- Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. Longman, London.
- Halliday, M. A. K. and Matthiessen, C. M. I. M. (2004). *An Introduction to Functional Grammar*. Edward Arnold, London, 3rd edition.
- Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C., editor, *WordNet: An Electronic Database and Some of its Applications*, pages 305–332. MIT Press.
- Janarthanam, S. and Lemon, O. (2009). Learning lexical alignment policies for generating referring expressions in spoken dialogue systems. In *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation*, pages 74–81, Morristown, NJ, USA.
- Kasper, R. (1989). SPL: A Sentence Plan Language for text generation. Technical report, USC/ISI.
- Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *ACL-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 704–710.
- Lemon, O. (2008). Adaptive Natural Language Generation in Dialogue using Reinforcement Learning. In *SemDial*.
- Levin, E., Pieraccini, R., and Eckert, W. (2000). A stochastic model of computer-human interaction for learning dialogue strategies. *IEEE Transactions on Speech and Audio Processing*, 8.
- Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialog. *Behavioral and Brain Sciences*, 27.
- Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.
- Rieser, V. and Lemon, O. (2009). Natural language generation as planning under uncertainty for spoken dialogue systems. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 683–691, Morristown, NJ, USA.
- Scott, D. and Moore, J. (2006). An NLG evaluation competition? eight reasons to be cautious. Technical report.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.
- Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70, Morristown, NJ, USA. Association for Computational Linguistics.
- Viethen, J. and Dale, R. (2006). Towards the evaluation of referring expression generation. In *In Proceedings of the 4th Australasian Language Technology Workshop*, pages 115–122.



# Tense and Aspect Assignment in Narrative Discourse

David K. Elson and Kathleen R. McKeown

Department of Computer Science

Columbia University

{delson,kathy}@cs.columbia.edu

## Abstract

We describe a method for assigning English tense and aspect in a system that realizes surface text for symbolically encoded narratives. Our testbed is an encoding interface in which propositions that are attached to a timeline must be realized from several temporal viewpoints. This involves a mapping from a semantic encoding of time to a set of tense/aspect permutations. The encoding tool realizes each permutation to give a readable, precise description of the narrative so that users can check whether they have correctly encoded actions and stative in the formal representation. Our method selects tenses and aspects for individual event intervals as well as subintervals (with multiple reference points), quoted and unquoted speech (which reassign the temporal focus) and modal events such as conditionals.

## 1 Introduction

Generation systems that communicate knowledge about time must select tense and aspect carefully in their surface realizations. An incorrect assignment can give the erroneous impression that a continuous action has ended, or that a previous state is the current reality. In this paper, we consider English tense and aspect in the generation of narrative discourse, where stative and actions occur over connected intervals.

We describe two contributions: first, a general application of theories of tense, aspect and interval logic to a generation context in which we map temporal relationships to specific tense/aspect selections. Second, we describe an implementation of this approach in an interactive environment with a basic sentence planner and realizer. The first result does not depend on the second.

The purpose of the system is to allow users who are naïve to linguistics and knowledge representa-

tion to create semantic encodings of short stories. To do this, they construct propositions (predicate-argument structures) through a graphical, menu-based interface, and assign them to intervals on a timeline. Figure 1 shows a session in which the user is encoding a fable of Aesop. The top-right panel shows the original fable, and the left-hand panel shows a graphical timeline with buttons for constructing new propositions at certain intervals. The left-hand and bottom-right panels contain automatically generated text of the encoded story, as the system understands it, from different points of view. Users rely on these realizations to check that they have assigned the formal connections correctly. The tenses and aspects of these sentences are a key component of this feedback. We describe the general purpose of the system, its data model, and the encoding methodology in a separate paper (Elson and McKeown, 2010).

The paper is organized as follows: After discussing related work in Section 2, we describe our method for selecting tense and aspect for single events in Section 3. Section 4 follows with more complex cases involving multiple events and shifts in temporal focus. We then discuss the results.

## 2 Related Work

There has been intense interest in the interpretation of tense and aspect into a formal understanding of the ordering and duration of events. This work has been in both linguistics (Dowty, 1979; Nerbonne, 1986; Vlach, 1993) and natural language understanding. Early systems investigated rule-based approaches to parsing the durations and orderings of events from the tenses and aspects of their verbs (Hinrichs, 1987; Weber, 1987; Song and Cohen, 1988; Passonneau, 1988). Allen (1984) and Steedman (1995) focus on distinguishing between achievements (when an event culminates in a result, such as *John builds a house*) and processes (such as walking). More

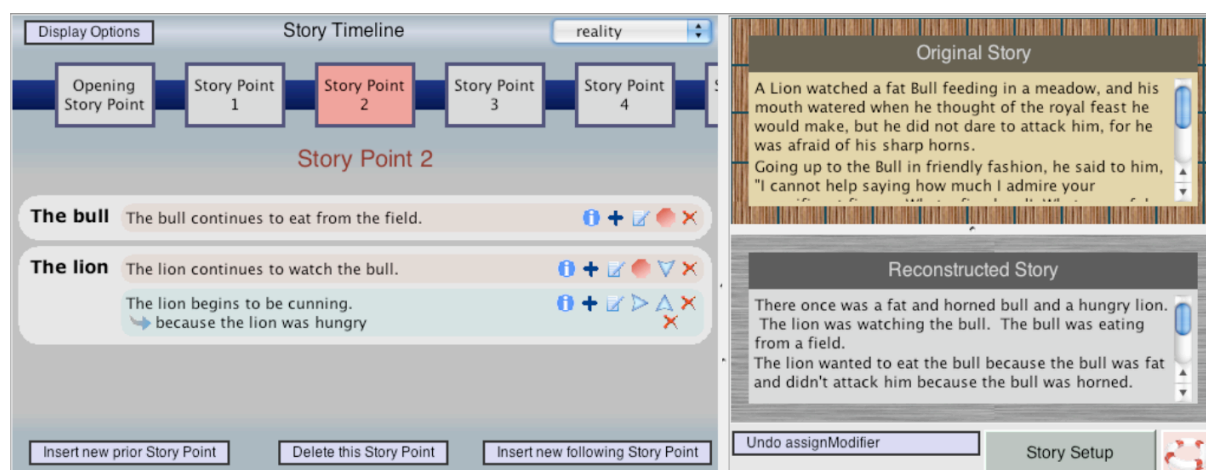


Figure 1: Screenshot of our story encoding interface.

recent work has centered on markup languages for complex temporal information (Mani, 2004) and corpus-based (statistical) models for predicting temporal relationships on unseen text (Mani et al., 2006; Lapata and Lascarides, 2006).

Our annotation interface required a simple realizer that could be easily integrated into an interactive, online encoding tool. We found that developing a custom realizer as a module to our Java-based system was preferable to integrating a large, general purpose system such as KPML/Nigel (Matthiessen and Bateman, 1991) or FUF/SURGE (Elhadad and Robin, 1996). These realizers, along with RealPro (Lavoie and Rambow, 1997), accept tense as a parameter, but do not calculate it from a semantic representation of overlapping time intervals such as ours (though the Nigel grammar can calculate tense from speech, event, and reference time orderings, discussed below). The statistically trained FERGUS (Chen et al., 2002) contrasts with our rule-based approach.

Dorr and Gaasterland (1995) and Grote (1998) focus on generating temporal connectives, such as *before*, based on the relative times and durations of two events; Gagnon and Lapalme (1996) focus on temporal adverbials (e.g., when to insert a known time of day for an event). By comparison, we extend our approach to cover direct/indirect speech and the subjunctive/conditional forms, which they do not report implementing. While our work focuses on English, Yang and Bateman (2009) describe a recent system for generating Chinese aspect expressions based on a time interval representation, using KPML as their surface realizer.

Several other projects run tangential to our in-

teractive narrative encoding project. Callaway and Lester’s STORYBOOK (2002) aims to improve fluency and discourse cohesion in realizing formally encoded narratives; Ligozat and Zock (1992) allow users to interactively construct sentences in various temporal scenarios through a graphical interface.

### 3 Expressing single events

#### 3.1 Temporal knowledge

The propositions that we aim to realize take the form of a predicate, one or more arguments, zero or more attached modifiers (either a negation operator or an adverbial, which is itself a proposition), and an assignment in time. Each argument is assigned a semantic role (such as Agent or Experiencer), and may include nouns (such as characters) or other propositions. In our implemented system, the set of predicates available to the annotator is adapted from the VerbNet (Kingsbury and Palmer, 2002) and WordNet (Fellbaum, 1998) linguistic databanks. These provide both durative actions and stative (Dowty, 1979); we will refer to both as *events* as they occur over intervals. For example, here are an action and a stative:

$$\text{walk}(\text{Mary}, \text{store}, 2, 6) \quad (1)$$

$$\text{hungry}(\text{Julia}, 1, \infty) \quad (2)$$

The latter two arguments in (1) refer to time states in a totally ordered sequence; Mary starts walking to the store at state 2 and finishes walking at state 6. (2) begins at state 1, but is unbounded (Julia never ceases being hungry). While this paper does not address the use of reference times



(such as equating a state to 6:00 or *yesterday*), this is an area of ongoing work.

(1) and (2), depending on the situation, can be realized in several aspects and tenses. We adapt and extend Reichenbach’s (1947) famous system of symbols for distinguishing between simple and progressive aspect. Reichenbach identifies three points that define the temporal position of the event: the event time  $E$ , the speech time  $S$ , and a reference time  $R$  which may or may not be indicated by a temporal adverbial. The total ordering between these times dictates the appropriate aspect. For example, the simple past *John laughed* has the relation  $S > E$ .  $R = E$  because there is no separate reference time involved. The past perfect *John had laughed [by the end of the play]* has the relation  $E < R < S$ , in that it describe “the past of the past”, with the nearer “past” being  $R$  (the end of the play).  $R$  can be seen as the temporal focus of the sentence.

As Reichenbach does not address events with intervals, we redefine  $E$  as the transition ( $E_1..E_2$ ) attached to the proposition (for example, (2,6) for Mary’s walk). This definition deliberately assumes that no event ever occurs over a single “instant” of time. The perception of an instantaneous event, when it is needed, is instead created by dilating  $R$  into a sufficiently large interval to contain the entire event, as in Dowty (1979).

We also distinguish between two generation modes: realizing the story as a complete discourse (*narration mode*), and describing the content of a single state or interval (*snapshot mode*). Our system supports both modes differently. Like most literary fiction, in discourse mode we realize the story as if all events occur before the speech time  $S$ . (We shall see that this does not preclude the use of the future tense in all cases.) In snapshot mode, speech time is concurrent with reference time so that the same events are realized as though they are happening “now.” The system uses this mode to allow annotators to inspect and edit what occurs at any point in the story. In Figure 1, for instance, the lion’s watching of the bull is realized as both a present, continuing event in snapshot mode (*the lion continues to watch the bull*) and narrated as a past, continuing event (*the lion was watching the bull*). In both cases, we aim to precisely translate the propositions and their temporal relationships into text, even if the results are not elegant rhetoric, so that annotators can see how they have

Diagram	Relations	Perspective
	$R < E_1$	Before
	$R = E_1$ $R < E_2$	Begin
	$E_1 < R$ $R < E_2$	During
	$R = E_2$ $R > E_1$	Finish
	$R > E_2$	After

Table 1: Perspective assignment for viewing an event from a reference state.

formally encoded the story. In the remainder of this section, we describe our method for assigning tenses and aspects to propositions such as these.

### 3.2 Reference state

In both snapshot and narration modes, we often need to render the events that occur at some reference state  $R$ . We would like to know, for instance, what is happening now, or what happened at 6:00 yesterday evening. The tense and aspect depend on the *perspective* of the reference state on the event, which can be bounded or unbounded. The two-step process for this scenario is to determine the correct perspective, then pick the tense and aspect class that best communicates it.

We define the set of possible perspectives to follow Allen (1983), who describes seven relationships between two intervals: before/after, meets/met by, overlaps/overlapped by, starts/started by, during/contains, finishes/finished by, and equals. Not all of these map to a relationship between a single reference *point* and an event interval. Table 1 maps each possible interaction between  $E$  and  $R$  to a perspective, for both bounded and unbounded events, including the defining relationships for each interaction. A diamond for  $E_1$  indicates *at or before*, i.e., the event is either anteriorly unbounded ( $E_1 = -\infty$ ) or beginning at a state prior to  $R$  and  $E_2$ . Similarly, a diamond for  $E_2$  indicates *at or after*.

Once the perspective is determined, covering Reichenbach’s  $E$  and  $R$ , speech time  $S$  is determined by the generation mode. Following the guidelines of Reichenbach and Dowty, we then assign a tense for each perspective/speech time per-

Perspective	Generation mode	English tense	System's construction	Example
<b>After</b>	Future Speech	Past perfect	<i>had</i> {PAST PARTICIPLE}	She had walked.
	Present Speech	Present perfect	<i>has/have</i> {PAST PARTICIPLE}	She has walked.
	Past Speech	Future perfect	<i>will have</i> {PAST PARTICIPLE}	She will have walked.
	Modal Infinitive		<i>to have</i> {PAST PARTICIPLE}	To have walked.
<b>Finish</b>	Future Speech	“Finished”	<i>stopped</i> {PROGRESSIVE}	She stopped walking.
	Present Speech	“Finishes”	<i>stops</i> {PROGRESSIVE}	She stops walking.
	Past Speech	“Will finish”	<i>will stop</i> {PROGRESSIVE}	She will stop walking.
	Modal Infinitive		<i>to stop</i> {PROGRESSIVE}	To stop walking.
<b>During</b>	Future Speech	Past progressive	<i>was/were</i> {PROGRESSIVE}	She was walking.
	Present Speech	Present progressive	<i>am/is/are</i> {PROGRESSIVE}	She is walking.
	Past Speech	Future progressive	<i>will be</i> {PROGRESSIVE}	She will be walking.
	Modal Infinitive		<i>to be</i> {PROGRESSIVE}	To be walking.
<b>During-After</b>	Future Speech	Past perfect progressive	<i>had been</i> {PROGRESSIVE}	She had been walking.
	Present Speech	Present perfect progressive	<i>has/have been</i> {PROGRESSIVE}	She has been walking.
	Past Speech	Future perfect progressive	<i>will have been</i> {PROGRESSIVE}	She will have been walking.
	Modal Infinitive		<i>to has/have been</i> {PROGRESSIVE}	To have been walking.
<b>Begin</b>	Future Speech	“Began”	<i>began</i> {INFINITIVE}	She began to walk.
	Present Speech	“Begins”	<i>begins</i> {INFINITIVE}	She begins to walk.
	Past Speech	“Will begin”	<i>will begin</i> {INFINITIVE}	She begins to walk.
	Modal Infinitive		<i>to begin</i> {PROGRESSIVE}	To begin walking.
<b>Contains</b>	Future Speech	Simple past	{SIMPLE PAST}	She walked.
	Present Speech	Simple present	{SIMPLE PRESENT}	She walks.
	Past speech	Simple future	<i>will</i> {INFINITIVE}	She will walk.
	Modal Infinitive		{INFINITIVE}	To walk.
<b>Before</b>	Future Speech	“Posterior”	<i>was/were going</i> {INFINITIVE}	She was going to walk.
	Present Speech	Future	<i>am/is/are going</i> {INFINITIVE}	She is going to walk.
	Past Speech	Future-of-future	<i>will be going</i> {INFINITIVE}	She will be going to walk.
	Modal Infinitive		<i>to be going</i> {INFINITIVE}	To be going to walk.

Table 2: Tense/aspect assignment and realizer constructions for describing an action event from a particular perspective and speech time. “Progressive” means “present participle.”

mutation in Table 2. Not all permutations map to actual English tenses. Narration mode is shown as *Future Speech*, in that  $S$  is in the future with respect to all events in the timeline. (This is the case even if  $E$  is unbounded, with  $E_2 = \infty$ .) Snapshot mode is realized as *Present Speech*, in that  $R = S$ . The fourth column indicates a syntactic construction with which our system realizes the permutation. Each is a sequence of tokens that are either cue words (*began*, *stopped*, etc.) or conjugations of the predicate’s verb. These constructions emphasize precision over fluency.

As we have noted, theorists have distinguished between “statives” that are descriptive (*John was hungry*), “achievement” actions that culminate in a state change (*John built the house*) and “activities” that are more continuous and divisible (*John read a book for an hour*) (Dowty, 1979). Prior work in temporal connectives has taken advantage of lexical information to determine the correct situation and assign aspect appropriately (Moens and

Steedman, 1988; Dorr and Gaasterland, 1995). In our case, we only distinguish between actions and statives, based on information from WordNet and VerbNet. We use a separate table for statives; it is similar to Table 2, except the constructions replace verb conjugations with insertions of *be*, *been*, *being*, *was*, *were*, *felt*, and so on (with the latter applying to affective states). We do not currently distinguish between achievements and activities in selecting tense and aspect, except that the annotator is tasked with “manually” indicating a new state when an event culminates in one (e.g., *The house was complete*). Recognizing an achievement action can benefit lexical choice (better to say *John finished building the house* than *John stopped*) and content selection for the discourse as a whole (the house’s completion is implied by *finished* and does not need to be stated).

To continue our running examples, suppose propositions (1) and (2) were viewed as a snapshot from state  $R = 2$ . Table 1 indicates *Begin*

Diagram	Relations	Perspective
	$R_1 \geq E_2$	After
	$R_1 > E_1$ $E_2 > R_1$ $R_2 > E_2$	Finish
	$R_1 \leq E_1$ $R_2 \geq E_2$	Contains
	$E_1 < R_1$ $E_2 > R_2$	During
	$R_1 < E_1$ $R_2 > E_1$ $E_2 > R_2$	Begin
	$E_1 \geq R_2$	Before

Table 3: Perspective assignment for describing an event from an assigned perspective.

to be the perspective for (1), since  $E_1 = R$ , and Table 2 calls for a “new” tense we have named *Begins* that means “begins at the present time.” When this tense’s construction is inserted into the overall syntax for *walk(Agent, Destination)*, which we derive from the VerbNet frame for *walk*, the result is *Mary begins to walk to the store*; similarly, (2) is realized as *Julia is hungry* via the *During* perspective. Narration mode uses past-tense verbs.

### 3.3 Reference interval

Just as events occur over intervals, rather than single points in time, so too can Reichenbach’s *R*. One may need to express what occurred when “Julia entered the room” (a non-instantaneous action) or “yesterday evening.” Our system allows annotators to view intervals in snapshot mode to get a sense of what happens over a certain time span.

The semantics of reference intervals have been studied as extensions to Reichenbach’s point approach. Dowty (1979, p.152), for example, posits that the progressive fits only if the reference interval is completely contained within the event interval. Following this, we construct an alternate lookup table (Table 3) for assigning the perspec-

Diagram	Relations	Perspective
	$E_2 > R_2$ $E_1 = -\infty$ $R_1 = -\infty$	During ( <i>a priori</i> )
	$R_2 > E_2$ $E_1 = -\infty$ $R_1 = -\infty$	After
	$R_1 > E_1$ $E_2 = \infty$ $R_2 = \infty$	Contains
	$E_1 > R_1$ $E_2 = \infty$ $R_2 = \infty$	Before

Table 4: Perspective assignment if event and reference intervals are unbounded in like directions.

tive of an event from a reference interval. Table 2 then applies in the same manner. In snapshot mode, the speech time *S* also occurs over an interval (namely, *R*), and Present Speech is still used. In narration mode, *S* is assumed to be a point following all event and reference intervals. In our running example, narrating the interval (1,7) results in *Mary walked to the store* and *Julia began to be hungry*, using the *Contains* and *Begin* perspectives respectively.

The notion of an *unbounded* reference interval, while unusual, corresponds to a typical perspective if the event is either bounded or unbounded in the opposite direction. These scenarios are illustrated in Table 3. Less intuitive are the cases where event and reference intervals are unbounded in the same direction. Perspective assignments for these instances are described in Table 4 and emphasize the bounded end of *R*. These situations occur rarely in this generation context.

### 3.4 Event Subintervals

Events are not always referred to in their entirety. We may wish to refer to the beginning, middle or end of an event, no matter when it occurs with respect to reference time. This invokes a second reference point (Comrie, 1985, p.128) in the same interval, delimiting a subinterval. Consider *John searches for his glasses* versus *John continues to search for his glasses*—both indicate an ongoing process, but the latter implies a subinterval during which time, we are expected to know, John was already looking for his glasses.

Our handling of subintervals falls along four alternatives that depend on the interval  $E_1..E_2$ , the reference *R* and the subinterval  $E'_1..E'_2$  of *E*, where  $E'_1 \geq E_1$  and  $E'_2 \leq E_2$ .

1. **During-After.** If  $E'$  is not a final subinterval of  $E$  ( $E'_2 < E_2$ ), and  $R = E'_2$  or  $R$  is a subinterval of  $E$  that is met by  $E'$  ( $R_1 = E'_2$ ), the perspective of  $E'$  is defined as *During-After*. In Table 2, this invokes the perfect-progressive tense. For example, viewing example (1) with  $E' = (2, 4)$  from  $R = 4$  in narration mode (Future Speech) would yield *Mary had been walking to the store*.
2. **Start.** Otherwise, if  $E'$  is an initial subinterval of  $E$  ( $E'_1 = E_1$  and  $E'_2 < E_2$ ), the perspective is defined as *Start*. These rows are omitted from Table 2 for space reasons, but the construction for this case reassigns the perspective to that between  $R$  and  $E'$ . Our realizer reassigns the verb predicate to *begin* (or *become* for statives) with a plan to render its only argument, the previous proposition, in the infinitive tense. For example, narrating (2) with  $E' = (1, 2)$  from  $R = 3$  would yield *Julia had become hungry*.
3. **Continue.** Otherwise, and similarly, if  $E$  strictly contains  $E'$  ( $E'_1 > E_1$  and  $E'_2 < E_2$ ), we assign the perspective *Continue*. To realize this, we reassign the perspective to that between  $R$  and  $E'$ , and reassign the verb predicate to *continue* (or *was still* for statives) with a plan to render its only argument, the original proposition, in the infinitive.
4. **End.** Otherwise, if  $E'$  is a final subinterval of  $E$  ( $E'_1 > E_1$  and  $E'_2 = E_2$ ), we assign the perspective *End*. To realize this, we reassign the perspective to that between  $R$  and  $E'$ , and reassign the verb predicate to *stop* (or *finish* for cumulative achievements). Similarly, the predicate's argument is the original proposition rendered in the infinitive.

#### 4 Alternate timelines and modalities

This section covers more complex situations involving *alternate timelines*—the feature of our representation by which a proposition in the main timeline can refer to a second frame of time. Other models of time have supported similar encapsulations (Crouch and Pulman, 1993; Mani and Pustejovsky, 2004). The alternate timeline can contain references to actual events or modal events (imagined, obligated, desired, planned, etc.) in the past the future with respect to its *point of attachment* on

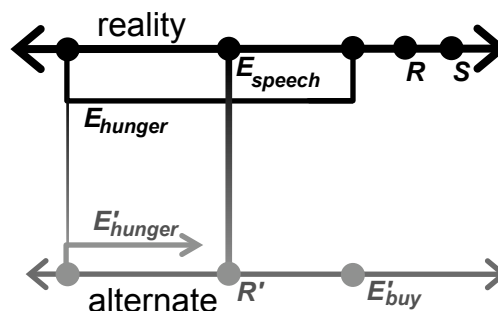


Figure 2: Schematic of a speech act attaching to an alternate timeline with a hypothetical action.  $R'$  and  $E_{speech}$  are attachment points.

the main timeline. This is primarily used in practice for modeling dialogue acts, but it can also be used to place real events at uncertain time states in the past (e.g., the present perfect is used in a reference story being encoded).

#### 4.1 Reassigning Temporal Focus

Ogihara (1995) describes dialogue acts involving changes in temporal focus as “double-access sentences.” We now consider a method for planning such sentences in such a way that the refocusing of time—the reassignment of  $R$  into a new context—is clear, even if it means changing tense and aspect mid-sentence. Suppose Mary were to declare that she would buy some eggs because of Julia’s hunger, but before she returned from the store, Julia filled up on snacks. If this speech act is described by a character later in the story, we need to carefully separate what is known to Mary at the time of her speech from what is later known at  $R$  by the teller of the episode. Mary sees her purchase of eggs as a possible future, even though it may have already happened by the point of retelling, and Mary does not know that Julia’s hunger is to end before long.

Following Hornstein’s treatment of these scenarios (Hornstein, 1990), we attach  $R'$ , the reference time for Mary’s statement (in an alternate timeline), to  $E_{speech}$ , the event of her speaking (in the main timeline). The act of buying eggs is a hypothetical event  $E'_{buy}$  that falls after  $R'$  on the alternate (modal) timeline.  $S$  is not reassigned here.

Figure 2 shows both timelines for this example. The main timeline is shown on top; Mary’s speech act is below. The attachment point on the main timeline is, in this case, the speech event  $E_{speech}$ ; the attachment point on an alternate timeline is al-

ways  $R'$ . The placement of  $R$ , the main reference point, is not affected by the alternate timeline. Real events, such as Julia's hunger, can be invoked in the alternate timeline (as drawn with a vertical line from  $E_{hunger}$  to an  $E'_{hunger}$  without an  $E'_2$  known to Mary) but they must preserve their order from the main timeline.

The tense assignment for the event intervals in the alternate timeline then proceeds as normal, with  $R'$  substituting for  $R$ . The hypothetical "buy" event is seen in *Before* perspective, but past tense (Future Speech), giving the "posterior" (future-of-a-past) tense. Julia's hunger is seen as *During* as per Table 1. Further, we assert that connectives such as *Because* do not alter  $R$  (or in this situation,  $R'$ ), and that the  $E'_{buy}$  is connected to  $E'_{hunger}$  with a causality edge. (Annotators can indicate connectives between events for causality, motivation and other features of narrative cohesion.)

The result is: *Mary had said that she was going to buy eggs because Julia was hungry.* The subordinate clause following *that* sees  $E'_{buy}$  in the future, and  $E'_{hunger}$  as ongoing rather than in the past. It is appropriately ambiguous in both the symbolic and rendered forms whether  $E'_{buy}$  occurs at all, and if so, whether it occurs before, during or after  $R$ . A discourse planner would have the responsibility of pointing out Mary's mistaken assumption about the duration of Julia's hunger.

We assign tense and aspect for quoted speech differently than for unquoted speech. Instead of holding  $S$  fixed,  $S'$  is assigned to  $R'$  at the attachment point of the alternate timeline (the "present time" for the speech act). If future hypothetical events are present, they invoke the Past Speech constructions in Table 2 that have not been used by either narration or snapshot mode. The content of the quoted speech then operates totally independently of the speech action, since both  $R'$  and  $S'$  are detached: *Mary said/says/was saying, "I am going to buy eggs because Julia is hungry."*

The focus of the sentence can be reassigned subsequently to deeper nested timelines as necessary (attaching an  $E'$  to  $R''$ , and so on). Although the above example uses subordinate clauses, we can use this nesting technique to construct composite tenses such as those enumerated by Halliday (1976). To this end, we conjugate the *Modal Infinitive* construction for each alternate timeline. Halliday's complex form "present in past in future in past in future," for instance (as in *will have been*

*going to have been taken*), can be generated with four timelines in a chain that invoke, in order and with Past Speech, the *After*, *Before*, *After* and *During* perspectives. There are four  $R$ s, all but the main one attached to a previous  $E$ .

## 4.2 Subjunctives and Conditionals

We finally consider tense and aspect in the case of subjunctive and conditional statements (if-thens), which appear in alternate timelines. The relationship between an *if* clause and a *then* clause is not the same as the relationship between two clauses joined by *because* or *when*. The *then* clause— or set of clauses— is predicated on the truth of the *if* clause. As linguists have noted (Hornstein, 1990, p.74), the *if* clause serves as an adverbial modifier, which has the effect of moving forward the reference point to the last of the *if* event intervals (provided that the *if* refers to a hypothetical future). Consider the example: *If John were to fly to Tokyo, he would have booked a hotel.* A correct model would place  $E'_{book}$  before  $E'_{fly}$  on the alternate timeline, with  $E'_{fly}$  as the *if*. Since *were to fly* is a hypothetical future,  $R' < E'_{fly}$ . During generation, we set  $R'$  to  $E'_{fly}$  after rendering *If John were to fly to Tokyo*, because we begin to assume that this event transpired. If  $R'$  is left unchanged, it may be erroneously left before  $E'_{book}$ : *Then he would be going to book a hotel.*

Our encoding interface allows users to mark one or more events in an alternate timeline as *if* events. If at least one event is marked, all *if* events are rendered in the subjunctive mood, and the remainder are rendered in the conditional. For the *if* clauses that follow  $R'$ ,  $S'$  and  $R'$  itself are reassigned to the interval for each clause in turn.  $R'$  and  $S'$  then remain at the latest *if* interval (if it is after the original  $R'$ ) for purposes of rendering the *then* clauses. In our surface realizer, auxiliary words *were* and *would* are combined with the Modal Infinitive constructions in Table 2 for events during or following the original attachment point.

As an example, consider an alternate timeline with two statives whose start and end points are the same: *Julia is hungry* and *Julia is unhappy*. The former is marked *if*. Semantically, we are saying that  $hungry(Julia) \rightarrow unhappy(Julia)$ . If  $R'$  were within these intervals, the rendering would be: *If Julia is hungry, then she is unhappy* (*Contains/Present Speech* for both clauses). If  $R'$  were prior to these intervals, the rendering

would be: *If Julia were to be hungry, then she would be unhappy*. This reassigns  $R'$  to  $E_{hungry}$ , using *were* as a futurate and *would* to indicate a conditional. Because  $R'$  and  $S'$  are set to  $E_{hungry}$ , the perspective on both clauses remains *Contains/Present Speech*. Finally, if both intervals are before  $R'$ , describing Julia's previous emotional states, we avoid shifting  $R'$  and  $S'$  backward: *If Julia had been hungry, then she had been unhappy* (*After* perspective, *Future Speech* for both statives).

The algorithm is the same for event intervals. Take (1) and a prior event where Mary runs out of eggs:

$$\text{runOut}(\text{Mary}, \text{eggs}, 0, 1) \quad (3)$$

Suppose they are in an alternate timeline with attachment point  $0'$  and (1) marked *if*. We begin by realizing Mary's walk as an *if* clause: *If Mary were to walk to the store*. We reassign  $R'$  to  $E_{walk}$ , (2,6), which diverts the perception of (3) from *Begins* to *After*: *She would have run out of eggs*. Conversely, suppose the conditional relationship were reversed, with (3) as the only *if* action. If the attachment point is  $3'$ , we realize (3) first in the *After* perspective, as  $R'$  does not shift backward: *If Mary had run out of eggs*. The remainder is rendered from the *During* perspective: *She would be walking to the store*. Note that in casual conversation, we might expect a speaker at  $R = 3$  to use the past simple: *If Mary ran out of eggs, she would be walking to the store*. In this case, the speaker is attaching the alternate timeline at a reference interval that subsumes (3), invoking the *Contains* perspective by casting a net around the past. We ask our annotators to select the best attachment point manually; automatically making this choice is beyond the scope of this paper.

## 5 Discussion

As we mentioned earlier, we are describing two separate methods with a modular relationship to one another. The first is an abstract mapping from a conceptual representation of time in a narrative, including interval and modal logic, to a set of 11 *perspectives*, including the 7 listed in Table 2 and the 4 introduced in Section 3.4. These 11 are crossed with three scenarios for speech time to give a total of 33 tense/aspect permutations. We also use an infinitive form for each perspective. One may take these results and map them from

other time representations with similar specifications.

The second result is a set of syntactic constructions for realizing these tenses at the surface level in our story encoding interface. Our focus here, as we have noted, is not fluency, but a surface-level rendering that reflects the relationships (and, at times, the ambiguities) present in the conceptual encoding. We consider changes in modality, such as an indicative reading as opposed to a conditional or subjunctive reading, to be at the level of the realizer and not another class of tenses.

We have run a collection project with our encoding interface and can report success in the tool's usability (Elson and McKeown, 2009). Two annotators each encoded 20 fables into the formal representation, with their only exposure to the semantic encodings being through the reference text generator (as in Figure 1). Both annotators became comfortable with the tool after a period of training; in surveys that they completed after each task, they gave Likert-scale usability scores of 4.25 and 4.30 (averaged over 20 tasks, with 5 meaning "easiest to use"). These scores are not specific to the generation component, but they suggest that annotators could derive satisfactory tenses from their semantic structures. The most frequently cited deficiency in the model in terms of time was the inability to assign reference times to states and intervals (such as *the next morning*).

## 6 Conclusion and Future Work

It has always been the goal in surface realization to generate sentences from a purely semantic representation. Our approach to the generation of tense and aspect from temporal intervals takes us closer to that goal. We have applied prior work in linguistics and interval theory and tested our approach in an interactive narrative encoding tool. Our method handles reference intervals and event intervals, bounded or unbounded, and extends into subintervals, modal events, conditionals, and direct and indirect speech where the temporal focus shifts.

In the future, we will investigate the limitations of the current model, including temporal adverbials (which explain the relationship between two events), reference times, habitual events, achievements, and discourse-level issues such as preventing ambiguity as to whether adjacent sentences occur sequentially (Nerbonne, 1986; Vlach, 1993).

## 7 Acknowledgments

This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-0935360. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

## References

- James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- James F. Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.
- Charles Callaway and James Lester. 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252.
- John Chen, Srinivas Bangalore, Owen Rambow, and Marilyn Walker. 2002. Towards automatic generation of natural language generation systems. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.
- Bernard Comrie. 1985. *Tense*. Cambridge University Press.
- Richard Crouch and Stephen Pulman. 1993. Time and modality in a natural language interface to a planning system. *Artificial Intelligence*, pages 265–304.
- Bonnie J. Dorr and Terry Gaasterland. 1995. Selecting tense, aspect, and connecting words in language generation. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada.
- David R. Dowty. 1979. *Word Meaning and Montague Grammar*. D. Reidel, Dordrecht.
- Michael Elhadad and Jacques Robin. 1996. An overview of surge: a reusable comprehensive syntactic realization component. In *INLG '96 Demonstrations and Posters*, pages 1–4, Brighton, UK. Eighth International Natural Language Generation Workshop.
- David K. Elson and Kathleen R. McKeown. 2009. A tool for deep semantic encoding of narrative texts. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 9–12, Suntec, Singapore.
- David K. Elson and Kathleen R. McKeown. 2010. Building a bank of semantically encoded narratives. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, Malta.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Michel Gagnon and Guy Lapalme. 1996. From conceptual time to linguistic time. *Computational Linguistics*, 22(1):91–127.
- Brigitte Grote. 1998. Representing temporal discourse markers for generation purposes. In *Proceedings of the Discourse Relations and Discourse Markers Workshop*, pages 22–28, Montreal, Canada.
- M.A.K. Halliday. 1976. The english verbal group. In G. R. Kress, editor, *Halliday: System and Function in Language*. Oxford University Press, London.
- Erhard W. Hinrichs. 1987. A compositional semantics of temporal expressions in english. In *Proceedings of the 25th Annual Conference of the Association for Computational Linguistics (ACL-87)*, Stanford, CA.
- Norbert Hornstein. 1990. *As Time Goes By: Tense and Universal Grammar*. MIT Press, Cambridge, MA.
- Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-02)*, Canary Islands, Spain.
- Mirella Lapata and Alex Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27:85–117.
- Benoit Lavoie and Owen Rambow. 1997. A fast and portable realizer for text generation systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC.
- Gerard Ligozat and Michael Zock. 1992. How to visualize time, tense and aspect? In *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92)*, pages 475–482, Nantes, France.
- Inderjeet Mani and James Pustejovsky. 2004. Temporal discourse models for narrative structure. In *Proceedings of the ACL Workshop on Discourse Annotation*, Barcelona, Spain.
- Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of COLING/ACL 2006*, pages 753–760, Sydney, Australia.
- Inderjeet Mani. 2004. Recent developments in temporal information extraction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP '03)*, pages 45–60, Borovets, Bulgaria.
- Christian M. I. M. Matthiessen and John A. Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Frances Pinter Publishers and St. Martin's Press, London and New York.

- Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.
- John Nerbonne. 1986. Reference time and time in narration. *Linguistics and Philosophy*, 9(1):83–95.
- Toshiyuki Ogihara. 1995. Double-access sentences and reference to states. *Natural Language Semantics*, 3:177–210.
- Rebecca Passonneau. 1988. A computational model of the semantics of tense and aspect. *Computational Linguistics*, 14(2):44–60.
- Hans Reichenbach. 1947. *Elements of Symbolic Logic*. MacMillan, London.
- Fei Song and Robin Cohen. 1988. The interpretation of temporal relations in narrative. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, St. Paul, Minnesota.
- Mark Steedman. 1995. Dynamic semantics for tense and aspect. In *The 1995 International Joint Conference on AI (IJCAI-95)*, Montreal, Quebec, Canada.
- Frank Vlach. 1993. Temporal adverbials, tenses and the perfect. *Linguistics and Philosophy*, 16(3):231–283.
- Bonnie Lynn Webber. 1987. The interpretation of tense in discourse. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-87)*, pages 147–154, Stanford, CA.
- Guowen Yang and John Bateman. 2009. The chinese aspect generation based on aspect selection functions. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP (ACL-IJCNLP 2009)*, Singapore.



# Textual Properties and Task Based Evaluation: Investigating the Role of Surface Properties, Structure and Content.

**Albert Gatt**

Institute of Linguistics  
University of Malta  
albert.gatt@um.edu.mt

**François Portet**

Laboratoire d'Informatique de Grenoble  
Grenoble Institute of Technology  
francois.portet@imag.fr

## Abstract

This paper investigates the relationship between the results of an extrinsic, task-based evaluation of an NLG system and various metrics measuring both surface and deep semantic textual properties, including relevance. The latter rely heavily on domain knowledge. We show that they correlate systematically with some measures of performance. The core argument of this paper is that more domain knowledge-based metrics shed more light on the relationship between deep semantic properties of a text and task performance.

## 1 Introduction

Evaluation methodology in NLG has generated a lot of interest. Some recent work suggested that the relationship between various intrinsic and extrinsic evaluation methods (Spärck-Jones and Galliers, 1996) is not straightforward (Reiter and Belz, 2009; Gatt and Belz, to appear), leading to some arguments for more domain-specific intrinsic metrics (Foster, 2008). One reason why these issues are important is that reliable intrinsic evaluation metrics that correlate with performance in an extrinsic, task-based setting can inform system development. Indeed, this is often the stated purpose of evaluation metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin and Hovy, 2003), which were originally characterised as evaluation ‘understudies’.

In this paper we take up these questions in the context of a knowledge-based NLG system, BT-45 (Portet et al., 2009), which summarises medical data for decision support purposes in a Neonatal Intensive Care Unit (NICU). Our extrinsic data comes from an experiment involving complex medical decision making based on automatically generated and human-authored texts (van der

Meulen et al., 2009). This gives us the opportunity to directly compare the textual characteristics of generated and human-written summaries and their relationship to decision-making performance. The present work uses data from an earlier study (Gatt and Portet, 2009), which presented some preliminary results along these lines for the system in question. We extend this work in a number of ways. Our principal aim is to test the validity not only of general-purpose metrics which measure surface properties of text, but also of metrics which make use of domain knowledge, in the sense that they attempt to relate the ‘deep semantics’ of the texts to extrinsic factors, based on an ontology for the BT-45 domain.

After an overview of related work in section 2, the BT-45 system, its domain ontology and the extrinsic evaluation are described in section 3. The ontology plays an important role in the evaluation metrics presented in Section 5. Finally, the evaluation of the methods is presented in Section 6, before discussing and concluding in Section 7.

## 2 Related Work

In NLG evaluation, extrinsic, task-based methods play a significant role (Reiter et al., 2003; Karasimos and Isard, 2004; Stock et al., 2007). Depending on the study design, these studies often leave open the question of precisely which aspects of a system (and of the text it generates) contribute to success or failure. Intrinsic NLG evaluations often involve ratings of text quality or responses to questionnaires (Lester and Porter, 1997; Callaway and Lester, 2002; Foster, 2008), with some studies using post-editing by human experts (Reiter et al., 2005). Automatically computed metrics exploiting corpora, such as BLEU, NIST and ROUGE, have mainly been used in evaluations of the coverage and quality of morphosyntactic realisers (Langkilde-Geary, 2002; Callaway, 2003), though they have recently also been used

for subtasks such as Referring Expression Generation (Gatt and Belz, to appear) as well as end-to-end weather forecasting systems (Reiter and Belz, 2009). The widespread use of these metrics in NLP partly rests on the fact that they are quick and cheap, but there is controversy about their reliability both in MT (Calliston-Burch et al., 2006) and summarisation (Dorr et al., 2005; Liu and Liu, 2008). As noted in Section 1, similar questions have been raised in NLG. One of the problems associated with these metrics is that they rely on the notion of a ‘gold standard’, which is not always precisely definable given multiple solutions to the same generation, summarisation or translation task. These observations underlie recent developments in Summarisation evaluation such as the Pyramid method (Nenkova and Passonneau, 2004), which in addition also emphasises *content* overlap with a set of reference summaries, rather than *n*-gram matches.

It is interesting to note that, with some exceptions (Foster, 2008), most of the methodological studies on intrinsic evaluation cited here have focused on ‘generic’ metrics (corpus-based automatic measures being foremost among them), none of which use domain knowledge to quantify those aspects of a text related to its content. There is some work in Summarisation that suggests that incorporating more knowledge improves results. For example, Yoo and Song (Yoo et al., 2007) used the Medical Subject Headings (MeSH) to construct graphs representing the high-level content of documents, which are then used to cluster documents by topic, each cluster being used to produce a summary. In (Plaza et al., 2009), the authors have proposed a summarisation method based on WordNet concepts and showed that this higher level representation improves the summarisation task.

The principal aim of this paper is to develop metrics with which to compare texts using domain knowledge – in the form of the ontology used in the BT-45 system – and to correlate results to human decision-making performance. The resulting metrics focus on aspects of content, structure and relevance that are shown to correlate meaningfully with task performance, in contrast to other, more surface-oriented ones (such as ROUGE).

### 3 The BT-45 System

BT-45 (Portet et al., 2009) was designed to gen-

erate a textual summary of 45 minutes of patient data in a Neonatal Intensive Care Unit (NICU), of the kind shown in Figure 1(a). The corresponding summary for the same data shown in Figure 1(b) is a two-step consensus summary written by two expert neonatologists. These two summaries correspond to two of the conditions in the task-based evaluation experiment described below.

In BT-45, summaries such as Figure 1(a) were generated from raw input data consisting of (a) physiological signals measured using sensors for various parameters (such as heart rate); and (b) discrete events logged by medical staff (e.g. drug administration). The system was based on a pipeline architecture which extends the standard NLG tasks such as document planning and microplanning with preliminary stages for data analysis and reasoning. The texts generated were descriptive, that is, they kept interpretation to a minimum (for example, the system did not make diagnoses). Nor were they generated with a bias towards specific problems or actions that could be considered desirable for a clinician to take in a particular context.

Every stage of the generation process made use of a domain-specific ontology of around 550 concepts, an excerpt of which is shown in Figure 1(c). The ontology classified objects of type EVENT and ENTITY into several subtypes; for example, a DRUG ADMINISTRATION is an INTERVENTION, which means it involves an agent and a patient. The ontology functioned as a repository of declarative knowledge, on the basis of which production rules were defined to support reasoning in order to make abstractions and to identify relations (such as causality) between events detected in the data based on their ontological class and their specific properties. In addition to the standard IS-A links, the ontology contains functional relationships which connect events to concepts representing physiological systems (such as the respiratory or cardiovascular systems); these are referred to as *functional concepts*. For example, in Figure 1(c), a FEED event is linked to NUTRITION, meaning that it is primarily relevant to the nutritional system. These links were included in the ontology following consultation with a senior neonatal consultant *after* the development of BT-45 was completed. Their inclusion was motivated by the knowledge-based evaluation metrics developed for the purposes of the present study, and discussed further

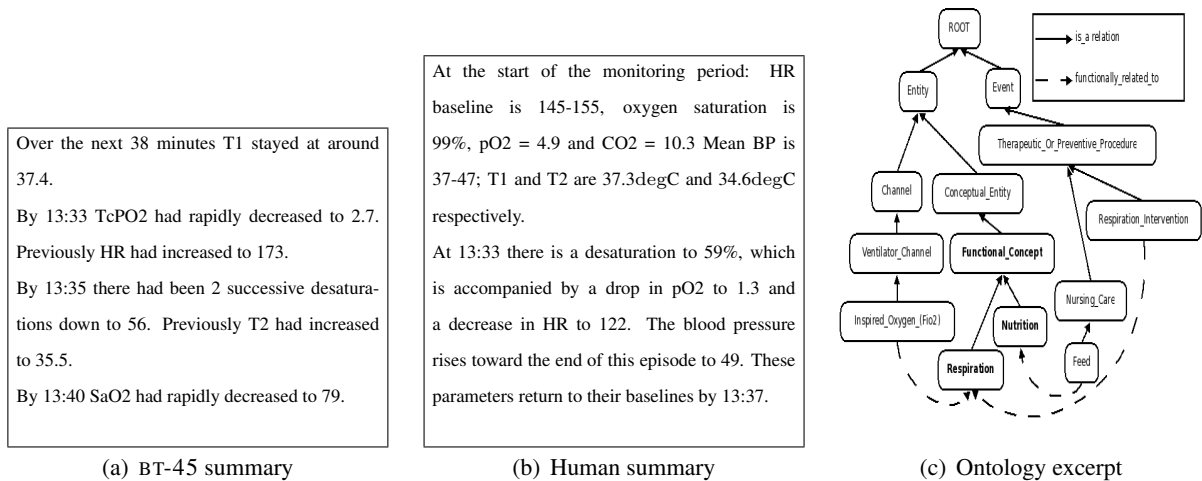


Figure 1: Excerpts from Human and BT-45 summaries, and ontology example.

in Section 5.

The task-based experiment to evaluate BT-45 was conducted off-ward and involved a group of 35 clinicians, who were exposed to 24 scenarios, each covering approximately 45 minutes of patient data, together with a short introductory text which gave some background about the patient. The patient data was then presented in one of three conditions: graphically using a time-series plot, and textually in the form of a consensus summary written by human experts (H; Figure 1(b)) and one generated automatically by BT-45(C; Figure 1(a)). Like the BT-45 texts, the H texts did not give interpretations or diagnoses and every effort was made not to bias a reader in favour of certain courses of action. A Latin Square design was used to ensure that each scenario was shown to an equal number of participants in each condition, while no participant saw the same scenario in more than one condition.

For each scenario, the task was to select one or more appropriate clinical actions from a predefined set of 18, one of which was ‘no action’. Selections had to be made within three minutes, after which the scenario timed out. The same choice of 18 actions was given in each scenario  $s$ , but for each one, two neonatal experts identified the subsets of appropriate ( $AP_s$ ), inappropriate/potentially harmful ( $INAP_s$ ) and neutral actions. One of the appropriate actions was also deemed to be the ‘target’, that is, the most important action to take. In three scenarios, the ‘target’ was ‘no action’. For each participant  $p$  and scenario  $s$ , the performance score  $P_s^p$  was based on the proportion  $P_{AP_s}$  of actions selected out of  $AP_s$ , and the proportion

$P_{INAP_s}$  selected out of the set of inappropriate actions  $INAP_s$ :  $P_s^p = P_{AP_s} - P_{INAP_s} \in [-1, 1]$ .

Overall, decision making in the H condition was better ( $P_s = .45^{SD=.10}$ ) than either C ( $P_s = .41^{SD=.13}$ ) or G ( $P_s = .40^{SD=.15}$ ). No significant difference was found between the latter two, but the H texts were significantly better than the C texts, as revealed in a by-subjects ANOVA ( $F(1, 31) = 5.266, p < 0.05$ ). We also performed a post-hoc analysis, comparing the proportions of appropriate actions selected,  $P_{AP}$  and that of inappropriate actions  $P_{INAP}$  in the H and C conditions across scenarios. In addition, we computed a different score  $SP_{AP}$ , defined as the proportion of appropriate actions selected by a participant within a scenario out of the total number of actions selected (effectively a measure of ‘precision’). A comparison between means for these three scores obtained across scenarios showed no significant differences.

In the analysis reported in Section 6, we compare our textual metrics to both the global score  $P$  as well as to these three other performance indicators. In various follow-up analyses (van der Meulen et al., 2009; Reiter et al., 2008), it was found that the three scenarios in which the target action was ‘no action’ may have misled some participants, insofar as this option was included among a set of other actions, some of which were themselves deemed appropriate or at least neutral (in the sense that they could be carried out without harming the patient). We shall therefore exclude these scenarios from our analyses.

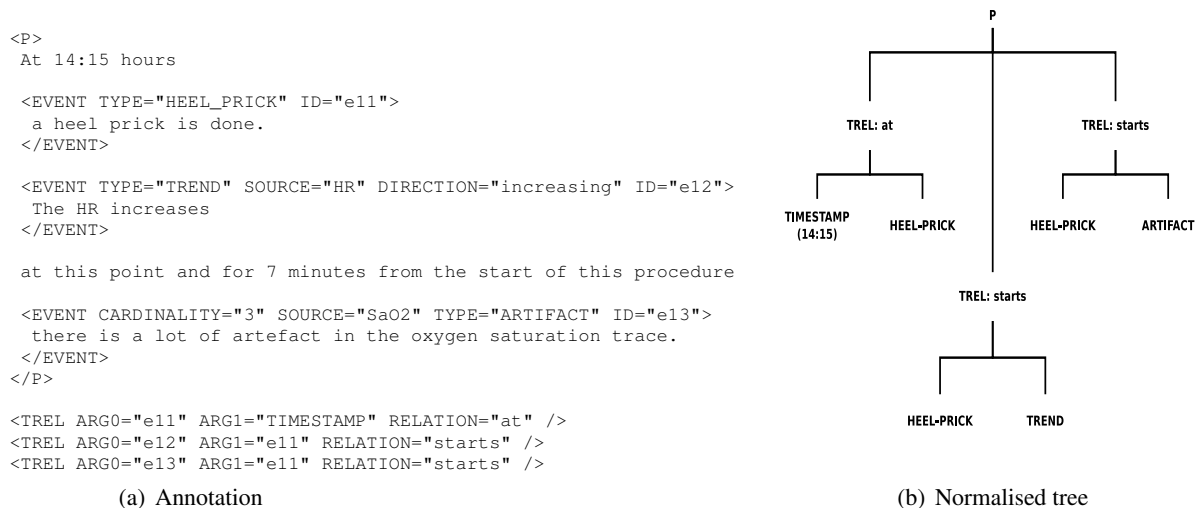


Figure 2: Fragment of an annotated summary and normalised tree representation.

## 4 Corpus Annotation

For this study, we annotated the H and C texts from our experiment using the ontology, in order to make both their semantic content and structure explicit. Figure 2(a) shows an excerpt from an annotated text. Every paragraph of the text is marked up explicitly. All segments of the text corresponding to an ontology EVENT are marked up with a TYPE (the name of the concept in the ontology) and other properties, such as DIRECTION and SOURCE in the case of trends in physiological parameters. The CARDINALITY attribute is used to indicate that a single text segment abstracts over several occurrences in the data; for example, the statement about artefacts in the example corresponds to three such episodes in the data.

In addition to events, the markup also includes separate nodes for all the temporal (TREL) and discourse (DREL) relations which are explicitly mentioned in the text, typically using adverbial or prepositional phrases or verbs of causality. Every TREL and DREL points to two arguments and has a RELATION attribute. In the case of a TREL, the value is one of the temporal relations defined by (Allen, 1983). For DRELS, values were restricted to CAUSE and CONTRAST (Mann and S.Thompson, 1988). One of the arguments of a TREL can be a timestamp, rather than an event. This is the case for the first sentence in the fragment, where event e11 is specified as having occurred at a specific time (*at 14:15*). By contrast, r4 is a relation between e11 and e12, where the RELATION is STARTS, indicating that the text

specifies that e11 is used by the author as the anchor point to specify the start of e12, as reflected by the expression *at this point*.

The markup provided the basis on which many of the metrics described in the following section were computed. Based on the annotation, we used a normalised structural representation of the texts as shown in Figure 2(b), consisting of PARAGRAPH (P) nodes which subsume events and relations. Relations dominate their event arguments. For example, the starts TREL holding between e12 and e11 is represented by a STARTS node subsuming the two events. In case an event is dominated by more than one relation (for example, it is temporally related to two events, as e11 is in Figure 2(a), we maintain the tree structure by creating two copies of the event, which are subsumed by the two relations. Thus, the normalised tree representation is a ‘compiled out’ version of the graph representing all events and their relations. The tree representation is better suited to our needs, given the complexity of comparing two graphs.

## 5 Metrics

The evaluation metrics used to score texts written by domain experts and those generated by the BT-45 system fall into three main classes, described below.

**Semantic content and structure** To compare both the content and the structure of texts, we used three measures. The first quantifies the number of EVENT nodes in an annotated text, defined as

$\sum_{e \in E} c$ , where  $E$  is the set of events mentioned, and  $c$  is the value of the `CARDINALITY` attribute of an event  $e \in E$ . Similarly, we computed the number of temporal (`TREL`) and discourse (`DREL`) relations mentioned in a text. We also used the Tree Edit Distance metric to compute the distance between the tree representations of the `H` and `C` texts (see Figure 2(b)). This measure computes the minimum number of node insertions, deletions and substitutions required to transform one tree into another and therefore takes into account not only the content (events and relations) but also its structural arrangement in the text. The edit distance between two trees is computed using the standard Levenshtein edit distance algorithm, computed over a string that represents the preorder traversal of the two trees, using a cost of 1 for insertions and deletions, and 2 for substitutions.

**N-gram overlap** As a measure of  $n$ -gram overlap, we use `ROUGE- $n$` , which measures simple  $n$ -gram overlap (in the present paper we use  $n = 4$ ). We also use `ROUGE-SU`, in which overlap is computed using skip-bigrams while also accounting for unigrams that a text has in common with its reference (in order to avoid bias against texts which share several unigrams but few skip bigrams).

**Domain-dependent relevance metrics** The metrics described so far make use of domain knowledge only to the extent that this is reflected in the textual markup. We now consider a family of metrics which are much more heavily reliant on domain-specific knowledge structures and reasoning. In our domain, the relevance of a text in a given experimental scenario  $s$  can be defined in terms of whether the events it mentions have some relationship to the appropriate clinical actions ( $AP_s$ ). We attempt to model some aspects of this using a weighting strategy and reasoning rules.

Recall from Section 3 that  $fc$ 's represent the various physiological systems to which an event or action can be related. Therefore, each event  $e$  mentioned in a text can be related to a set of possible actions using the functional concepts  $fc(e)$  to which that event is linked in the ontology. Let  $E_{s,t}$  be the set of events mentioned in text  $t$  for scenario  $s$ . An event  $e \in E_{s,t}$  *references* an action  $a$  iff  $FC(e) \cap FC(a) \neq \emptyset$ . Our hypothesis is that an appropriate action is more likely to be taken if

there are events which reference it in the text – that is, if the text mentions things which are directly or indirectly *relevant* to the action. For instance, if a text mentions events related to the `RESPIRATION`  $fc$ , a clinician might be more likely to make a decision to manage a patient's ventilation support. It is worth emphasising that, since both the `BT-45` and human-authored texts were descriptive and were not written or generated with the appropriate actions in mind, the hypothesis that the relevance of the content to the appropriate actions might increase the likelihood of these actions being chosen is far from a foregone conclusion.

Part of the novelty in this way of quantifying relevance lies in its use of the knowledge (i.e. the ontology) that is already available to the system, rather than asking human experts to rate the relevance of a text, a time-consuming process which could be subject to experts' personal biases. However, this way of conceptualising relevance generates links to too many actions for one event. It is often the case that an event, through its association with a functional concept, references more than one action, but not all of these are appropriate. For example, a change in oxygen saturation can be related to `RESPIRATION`, which itself is related to several respiration-related actions in a scenario, only some of which are appropriate. Clearly, relevance depends not only on a physiological connection between an event and a physiological system (functional concept), but also on the *context*, that is, the other events and their relative importance in a given scenario. Another factor that needs to be taken into account is the overall probability of an action. Some actions are performed routinely, while others tend to be associated with emergencies (e.g. a nappy change is much more frequent over all than resuscitating a patient). This means that some actions – even appropriate ones – may have been less likely to be selected even though they were referenced by the text and were appropriate.

We prune unwarranted connections between events and actions by taking into account (a) a patient's current status (described in the text and in the background information given to experimental participants); (b) the fact that some actions have much higher prior probabilities than others because they are performed more routinely; (c) the fact that some events may be more important than others (e.g. resuscitation is much more important

than a nappy change). Based on this, we define the *weight* of an action  $a$  as follows:

$$W_a = \frac{\sum_{e \in E} \frac{Pr(a) * e.importance}{\sum_{a \in A_e} Pr(a)}}{\sum_{e \in E} e.importance} \quad (1)$$

Where  $E$  is the set of events in the text,  $A_e$  the set of actions related to event  $e$ ,  $e.importance \in \mathbb{N}^+$  the importance of the event  $e$  and  $Pr(a)$  the prior probability of action  $a$ . All weights are normalised so that the following inequalities hold:

$$\sum_{a \in A_e} \frac{Pr(a) * e.importance}{\sum_{a \in A_e} Pr(a)} = e.importance \quad (2)$$

$$\sum_{a \in A} W_a = 1 \quad (3)$$

where  $A$  is the set of all possible actions. The idea is that an event  $e$  makes some contribution (possibly 0) to the relevance of some actions  $A_e$ , and the total weight of the event is distributed among all actions related to it using (a) the prior probability  $Pr(a)$  of each action (the most frequent action will have more weight) and (b) the importance of the event. At the end of the process each action would be assigned a score representing the accumulated weights of the events, which is then normalised, so that  $\sum_{a \in A} W_a = 1$ .

The prior probability in the equation is meant to reflect our earlier observation that clinical actions differ in the frequency with which they are performed and this may bias their selection. Priors were computed using maximum likelihood estimates from a large database containing exhaustive annotations of clinical actions recorded by an on-site research nurse over a period of 4 months in a NICU, which contains a total of 43,889 records of actions (Hunter et al., 2003).

The importance value in equation (1) is meant to reflect the fact that events in the text do not attract the attention of a reader to the same extent, since they do not have the same degree of ‘severity’ or ‘surprise’. We operationalise this by identifying the superconcepts in the ontology (PATHOLOGICAL-FUNCTION, DISORDER, SURGICAL-INTERVENTION, etc.) which could be thought of as representing ‘drastic’ occurrences. To these we added the concept of a TREND which corresponds to a change in a physiological parameter (such as an increase in heart rate), based on the rationale that the primary aim of NICU staff is to keep a patient stable, so that any physiological instability warrants an intervention. The importance

of events subsumed by these superconcepts was then set to be three times that of ‘normal’ events.

Finally, we apply knowledge-based rules to prune the number of actions  $A_e$  related to an event  $e$ . As an example, a decision to intubate a baby depends not only on events in the text which reference this action, but also on whether the baby is already intubated. This can be assessed by checking whether s/he is on CMV (a type of ventilation which is only used after intubation). The rule is represented as  $INTUBATE \rightarrow \neg on(baby, CMV)$ . Although such rules are extremely rough, they do help to prune inconsistencies.

Two scores were computed for both human and computer texts using equation (1).  $REL_{s,t}$  is the sum of weights of actions referenced in a text  $t$  for scenario  $s$  which are appropriate:  $REL_{s,t} = \sum_{a \in A_{ap}} W_a$ . Conversely,  $IRREL_{s,t}$  quantifies the weights of actions referenced in  $t$  for scenario  $s$  which are inappropriate:  $IRREL_{s,t} = \sum_{a \in A_{inap}} W_a$ .

## 6 Results

In what follows, we report two-tailed Pearson’s  $r$  correlations to compare our metrics to the three performance measures discussed in Section 3:  $P$ , the global performance score;  $P_{APP}$  and  $P_{INAPP}$ , the proportion of appropriate (resp. inappropriate) actions selected from the subsets of in/appropriate (resp. inappropriate) actions in a scenario; and  $SP_{APP}$ , the proportion of appropriate actions selected by a participant out of the set of actions selected. The last three are included because they shed light more directly on the extent to which experimental participants chose correctly or incorrectly. In case a metric measures similarity or difference between texts, the correlation reported is with the *difference* between the H scores and the C scores. Where relevant, we also report correlations with the *absolute* mean performance scores within the H and/or C conditions. Correlations exclude the three scenarios which had ‘no action’ as the target appropriate action, though where relevant, we will indicate whether the correlations change when these scenarios are also included.

### 6.1 Content and Structure

Overall, the C texts mentioned significantly fewer events than the H texts ( $t_{20} = 2.44, p = .05$ ), and also mentioned fewer temporal and discourse relations explicitly ( $t_{20} = 3.70, p < .05$ ). In

	$P$ (H-C)	$P_{APP}$ (H-C)	$SP_{APP}$ (H-C)	$P_{INAP}$ (H-C)
Events (H-C)	.43 $\diamond$	.42 $\clubsuit$	.02	-.09
Relations (H-C)	.34	.30	0	-.15
Tree Edit	.36	.33	.09	-.14

Table 1: Correlations between performance differences and content/structure measures.  $\diamond$  significant at  $p = .05$ ;  $\clubsuit$  approaches significance at  $p = .06$

the case of the H texts, the number of events and relations did not correlate significantly with any of the performance scores. In the case of the C texts, the number of relations mentioned was significantly negatively correlated to  $P_{INAP}$  ( $r = -.49, p < .05$ ), and positively correlated to  $SP_{APP}$  ( $r = .7, p < .001$ ). This suggests that temporal and discourse relations made texts more understandable and resulted in more appropriate actions being taken. More unexpectedly, the number of events mentioned was negatively correlated to  $P_{APP}$  ( $r = -.53, p < .05$ ) and to  $P$  ( $r = -.5, p < .05$ ). This may have been due to the C texts mentioning a number of events that were relatively unimportant and/or irrelevant to the appropriate actions.

Table 1 displays correlations between performance differences between H and C, and differences in number of events and relations, as well as Tree Edit Distance. The positive correlation between the number of events mentioned and  $P$  suggests that a larger amount of content in the H texts is partially responsible for the difference in decision-making accuracy by experimental participants. This is further supported by the fact that the correlation with the difference in  $P_{APP}$  approaches significance. It is worth noting that none of these correlations are significant when means from the three ‘no action’ scenarios are included in the computation. This further supports our earlier conclusion that these three scenarios are outliers. Somewhat surprisingly, Tree Edit Distance does not correlate significantly with any of the performance differences, though the correlations go in the expected directions (positive in the case of  $P$ ,  $SP_{APP}$  and  $P_{APP}$ , negative in the case of  $P_{INAP}$ ). This may be due to the high variance in the Edit Distance scores (mean: 66.5; SD: 34.8).

Overall, these results show that differences in both content and structure made the H texts superior and human texts did a much better job at explicitly relating events or situating them in time, which is crucial for comprehension and correct decision-making. This point has previously been

	Absolute Scores (C)				Differences (H-C)			
	$P$	$P_{AP}$	$P_{INAP}$	$SP_{AP}$	$P$	$P_{AP}$	$P_{INAP}$	$SP_{AP}$
R-4	.33	.38	.2	-.03	-.19	-.2	-.01	-.1
R-SU	-.03	-.02	.05	-.31	.04	.01	-.1	.13

Table 2: Correlations between ROUGE and performance scores in the C condition.  $\diamond$  significant at  $p = .05$ .

made in relation to the same data on the basis of a qualitative study (Reiter et al., 2008).

## 6.2 N-gram Overlap

Correlations with ROUGE-4 and ROUGE-SU are shown in Table 2 both for absolute performance scores on the C texts, and for the differences between H and C. This is because ROUGE can be interpreted in two ways: on the one hand, it measures the ‘quality’ of C texts relative to the reference human texts; on the other it also indicates similarity between C and H.

There are no significant correlations between ROUGE and any of our performance measures. Although this leaves open the question of whether a different set of performance measures, or a different experiment, would evince a more systematic covariation, the results suggest that it is not surface similarity (to the extent that this is measured by ROUGE) that is contributing to better decision making. It is however worth noting that some correlations with ROUGE-4, namely those involving  $P$  and  $P_{APP}$ , do turn out significant when the ‘no action’ scenarios are included. This turns out to be solely due to one of the ‘no action’ scenarios, which had a much higher ROUGE-4 score than the others, possibly because the corresponding human text was comparatively brief and the number of events mentioned in the two texts was roughly equal (11 for the C text, 12 for the H text).

## 6.3 Knowledge Based Relevance Metrics

Finally, we compare our knowledge-based measures of the relevance of the content to appropriate actions (REL) and to inappropriate actions (IR-REL). The correlations between each measure and

	Human (H)				BT-45 (C)			
	$P$	$P_{AP}$	$P_{INAP}$	$SP_{AP}$	$P$	$P_{AP}$	$P_{INAP}$	$SP_{AP}$
REL	.14	.11	-.14	.60 $\diamond$	.33	.24	-.49 $\diamond$	.7 $\diamond$
IRREL	-.25	-.22	.1	-.56 $\diamond$	-.34	-.26	.43	-.62 $\diamond$

Table 3: Correlations between knowledge-based relevance scores and absolute performance scores in the C and H conditions.  $\diamond$  significant at  $p \leq .05$ .

the absolute performance scores in each condition are displayed in Table 3.

The absolute scores in Table 3 show that both REL and IRREL are significantly correlated to  $SP_{APP}$ , the proportion of appropriate actions out of the actions selected by participants. The correlations are in the expected direction: there is a strong tendency for participants to choose more appropriate actions when REL is high, and the reverse is true for IRREL. In the case of the C texts, there is also a negative correlation (as expected) between REL and  $P_{INAP}$ , though this is the only one that reaches significance with this variable. It therefore appears that the knowledge-based relevance measures evince a meaningful relationship with at least some of the more ‘direct’ measures of performance (those assessing the relative preference of participants for appropriate actions based on a textual summary), though not with the global preference score  $P$ . One possible reason for the low correlations with the latter is that the two measures attempt to quantify directly the relevance of the content units in a text to in/appropriate courses of action; hence, they have a more direct relationship to measures of proportions of the courses of actions chosen.

## 7 Discussion and Conclusions

We conclude this paper with some observations about the relative merit of different measures of textual characteristics. ‘Standard’, surface-based measures such as (ROUGE) do not display any systematic relationship with our extrinsic measures of performance, recalling similar observations in the NLG literature (Gatt and Belz, to appear) and in MT and Summarisation (Calliston-Burch et al., 2006; Dorr et al., 2005). Some authors have also reported that ROUGE does not correlate well with human judgements of NLG texts (Reiter and Belz, 2009). On the other hand, we do find some evidence that the amount of content in texts, and the extent to which they explicitly relate content elements temporally and rhetorically, may impact decision-making. The significant correlations ob-

served between the number of relations in a text and the extrinsic measures are worth emphasising, as they suggest a significant role not only for content, but also rhetorical and temporal structure, something that many metrics do not take into account.

Perhaps the most important contribution of this paper has been to emphasise knowledge-based aspects of textual evaluation, not only by measuring content units and structure, but also by developing a motivated *relevance* metric, the crucial assumption being that the utility of a summary is contingent on its managing to convey information that will motivate a reader to take the ‘right’ course of action. The strong correlations between the relevance measures and the extent to which people chose the correct actions (or more accurately, chose *more* correct actions) vindicates this assumption.

Some of the correlations which turned out not to be significant may be due to ‘noise’ in the data, in particular, high variance in the performance scores (as suggested by the standard deviations for  $P$  given in Section 3). They therefore do not warrant the conclusion that *no* relationship exists between a particular measure and extrinsic task performance; nevertheless, where other studies have noted similar gaps, the trends in question may be systematic and general. This, however, can only be ascertained in further follow-up studies.

This paper has investigated the relationship between a number of intrinsic measures of text quality and decision-making performance based on an external task. Emphasis was placed on metrics that quantify aspects of semantics, relevance and structure. We have also compared generated texts to their human-authored counterparts to identify differences which can motivate further system improvements. Future work will focus on further exploring metrics that reflect the relevance of a text, as well as the role of temporal and discourse structure in conveying the intended meaning.



## References

- J. F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Charles B. Callaway and James C. Lester. 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252.
- C. B. Callaway. 2003. Evaluating coverage for large symbolic NLG grammars. In *Proc. IJCAI'03*.
- C. Calliston-Burch, M. Osborne, and P. Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proc. EACL'06*.
- B. J. Dorr, C. Monz, S. President, R. Schwartz, and D. Zajic. 2005. A methodology for extrinsic evaluation of text summarization: Does ROUGE correlate? In *Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures*.
- M.E. Foster. 2008. Automated metrics that agree with human judgements on generated output for an embodied conversational agent. In *Proc. INLG'08*.
- A. Gatt and A. Belz. to appear. Introducing shared task evaluation to NLG: The TUNA shared task evaluation challenges. In E. Kraemer and M. Theune, editors, *Empirical Methods in Natural Language Generation*. Springer.
- A. Gatt and F. Portet. 2009. Text content and task performance in the evaluation of a natural language generation system. In *Proc. RANLP'09*.
- J. Hunter, G. Ewing, L. Ferguson, Y. Freer, R. Logie, P. McCue, and N. McIntosh. 2003. The NEONATE database. In *Proc. IDAMAP'03*.
- A. Karasimos and A. Isard. 2004. Multilingual evaluation of a natural language generation system. In *Proc. LREC'04*.
- I. Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. INLG'02*.
- J.C. Lester and B.W. Porter. 1997. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.
- C-Y Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. in. In *Proc. of HLT-NAACL'03*.
- F. Liu and Y. Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. In *Proc. ACL'08*.
- W. C. Mann and S. Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organisation. *Text*, 8(3):243–281.
- A. Nenkova and R. Passonneau. 2004. Evaluating content selection in summarisation: The Pyramid method. In *Proc. NAACL-HLT'04*.
- S. Papineni, T. Roukos, W. Ward, and W. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. ACL'02*.
- L. Plaza, A. Díaz, and P. Gervás P. 2009. Automatic summarization of news using wordnet concept graphs. best paper award. In *Proc. IADIS'09*.
- F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7–8):789–816.
- E. Reiter and A. Belz. 2009. An investigation into the validity of some metrics for automatically evaluating Natural Language Generation systems. *Computational Linguistics*, 35(4):529–558.
- E. Reiter, R. Robertson, and L. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144:41–58.
- E. Reiter, S. Sripada, J. Hunter, J. Yu, and I. Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137–169.
- E. Reiter, A. Gatt, F. Portet, and M. van der Meulen. 2008. The importance of narrative and other lessons from an evaluation of an NLG system that summarises clinical data. In *Proc. INLG'08*.
- K. Spärck-Jones and J. R. Galliers. 1996. *Evaluating natural language processing systems: An analysis and review*. Springer, Berlin.
- O. Stock, M. Zancanaro, P. Busetta, C. Callaway, A. Krueger, M. Kruppa, T. Kuflik, E. Not, and C. Rocchi. 2007. Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction*, 17(3):257–304.
- M. van der Meulen, R. H. Logie, Y. Freer, C. Sykes, N. McIntosh, and J. Hunter. 2009. When a graph is poorer than 100 words. *Applied Cognitive Psychology*, 24(1):77–89.
- I. Yoo, X. Hu, and I-Y Song. 2007. A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. *BMC Bioinformatics*, 8(9).



# Situated Reference in a Hybrid Human-Robot Interaction System

Manuel Giuliani<sup>1</sup> and Mary Ellen Foster<sup>2</sup> and Amy Isard<sup>3</sup>  
Colin Matheson<sup>3</sup> and Jon Oberlander<sup>3</sup> and Alois Knoll<sup>1</sup>

<sup>1</sup>Informatik VI: Robotics and Embedded Systems, Technische Universität München

<sup>2</sup>School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh

<sup>3</sup>Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh

## Abstract

We present the situated reference generation module of a hybrid human-robot interaction system that collaborates with a human user in assembling target objects from a wooden toy construction set. The system contains a sub-symbolic goal inference system which is able to detect the goals and errors of humans by analysing their verbal and non-verbal behaviour. The dialogue manager and reference generation components then use situated references to explain the errors to the human users and provide solution strategies. We describe a user study comparing the results from subjects who heard constant references to those who heard references generated by an adaptive process. There was no difference in the objective results across the two groups, but the subjects in the adaptive condition gave higher subjective ratings to the robot's abilities as a conversational partner. An analysis of the objective and subjective results found that the main predictors of subjective user satisfaction were the user's performance at the assembly task and the number of times they had to ask for instructions to be repeated.

## 1 Introduction

When two humans jointly carry out a mutual task for which both know the plan—for example, assembling a new shelf—it frequently happens that one makes an error, and the other has to assist and to explain what the error was and how it can be solved. Humans are skilled at spotting errors committed by another, as well as errors which they made themselves. Recent neurological studies have shown that error monitoring—i.e., observing the errors made by oneself or by others—

plays an important role in joint activity. For example, Bekkering et al. (2009) have demonstrated that humans show the same brain activation patterns when they make an error themselves and when they observe someone else making an error.

In this paper, we describe a human-robot interaction (HRI) system that is able both to analyse the actions and the utterances of a human partner to determine if the human made an error in the assembly plan, and to explain to the human what went wrong and what to do to solve the problem. This robot combines approaches from sub-symbolic processing and symbolic reasoning in a hybrid architecture based on that described in Foster et al. (2008b).

During the construction process, it is frequently necessary to refer to an object which is being used to assemble the finished product, choosing an unambiguous reference to distinguish the object from the others available. The classic reference generation algorithm, on which most subsequent implementations are based, is the incremental algorithm of Dale and Reiter (1995), which selects a set of attributes of a target object to single it out from a set of distractor objects. In real-world tasks, the speaker and hearer often have more context in common than just the knowledge of object attributes, and several extensions have been proposed, dealing with visual and discourse salience (Kelleher and Kruijff, 2006) and the ability to produce multimodal references including actions such as pointing (van der Sluis, 2005; Kranstedt and Wachsmuth, 2005).

Foster et al. (2008a) noted another type of multimodal reference which is particularly useful in embodied, task-based contexts: *haptic-ostensive* reference, in which an object is referred to as it is being manipulated by the speaker. Manipulating an object, which must be done in any case as part of the task, also makes an object more salient and therefore affords linguistic references that in-



Figure 1: The dialogue robot

indicate the increased accessibility of the referent. This type of reference is similar to the *placing-for* actions noted by Clark (1996).

An initial approach for generating referring expressions that make use of haptic-ostensive reference was described in (Foster et al., 2009a). With this system, a study was conducted comparing the new reference strategy to the basic Dale and Reiter incremental algorithm. Naïve users reported that it was significantly easier to understand the instructions given by the robot when it used references generated by the more sophisticated algorithm. In this paper, we perform a similar experiment, but making use of a more capable human-robot interaction system and a more complete process for generating situated references.

## 2 Hybrid Human-Robot Dialogue System

The experiment described in this paper makes use of a hybrid human-robot dialogue system which supports multimodal human-robot collaboration on a joint construction task. The robot (Figure 1) has a pair of manipulator arms with grippers, mounted in a position to resemble human arms, and an animatronic talking head (van Breemen, 2005) capable of producing facial expressions, rigid head motion, and lip-synchronised synthesised speech. The subject and the robot work together to assemble wooden construction toys on a common workspace, coordinating their actions

through speech (English or German), gestures, and facial expressions.

The robot can pick up and move objects in the workspace and perform simple assembly tasks. In the scenario considered here, both of the participants know the assembly plan and jointly execute it. The robot assists the human, explains necessary assembly steps in case the human makes an error, and offers pieces as required. The workspace is divided into two areas—one belonging to the robot and one to the human—to make joint action necessary for task success.

The system has components which use both sub-symbolic and symbolic processing. It includes a goal inference module based on dynamic neural fields (Erlhagen and Bicho, 2006; Bicho et al., 2009), which selects the robot's next actions based on the human user's actions and utterances. Given a particular assembly plan and the knowledge of which objects the user has picked up, this module can determine when the user has made an error. The system also incorporates a dialogue manager based on the TrindiKit dialogue management toolkit (Larsson and Traum, 2000), which implements the information-state based approach to dialogue management. This unique combination of abilities means that when the robot detects that its human partner has made an error—for example, picking up or requesting an assembly piece that is not needed in the current step of the building plan—it can explain to the human what the error was and what can be done to correct the mistake—for example by picking up or indicating the correct assembly piece.

Messages from all of the system's input channels (speech, object recognition, and gesture recognition) are processed and combined by a multimodal fusion component based on (Giuliani and Knoll, 2008), which is the link between the symbolic and the sub-symbolic parts of the system. The fusion component then communicates with the goal inference module, which calculates the next action instructions for the robot and also determines if the user made an error. From there, fusion combines the information from goal inference with the input data and sends unified hypotheses to the dialogue manager.

When it receives the fusion hypotheses, the dialogue manager uses the dialogue history and the physical and task context to choose a response. It then sends a high-level specification of the de-

- 
1. **System** First we will build a windmill.
  2. **User** Okay.
  3. **User** {*picks up a yellow cube, unnecessary piece for a windmill*}
  4. **System** You don't need a yellow cube to build a windmill.
  5. **System** To build a windmill, you first need to build a tower.
  6. **System** [*picking up and holding out red cube*] To build the tower, insert the green bolt through the end of this red cube and screw it into the blue cube.
  7. **User** [*takes cube, performs action*] Okay.
- 

Figure 2: Sample human-robot dialogue, showing adaptively-generated situated references

sired response to the output planner, which in turn sends commands to each output channel: linguistic content (including multimodal referring expressions), facial expressions and gaze behaviours of the talking head, and actions of the robot manipulators. The linguistic outputs are realised using the OpenCCG surface realiser (White, 2006).

### 3 Reference Generation

In this system, two strategies were implemented for generating references to objects in the world: a constant version that uses only the basic incremental algorithm (Dale and Reiter, 1995) to select properties, and an adaptive version that uses more of the physical, dialogue and task context to help select the references. The constant system can produce a definite or indefinite reference, and the most appropriate combination of attributes according to the incremental algorithm. The adaptive system also generates pronominal and deictic references, and introduces the concept of multiple types of distractor sets depending on context.

Figure 2 shows a fragment of a sample interaction in which the user picks up an incorrect piece: the robot detects the error and describes the correct assembly procedure. The underlined references show the range of output produced by the adaptive reference generation module; for the constant system, the references would all have been “the red cube”. The algorithms used by the adaptive reference generation module are described below.

#### 3.1 Reference Algorithm

The module stores a history of the referring expressions spoken by both the system and the user, and uses these together with distractor sets to select referring expressions. In this domain there are two types of objects which we need to refer to: concrete objects in the world (everything which is on the table, or in the robot's or user's hand), and objects which do not yet exist, but are in the process of being created. For non-existent objects we do not build a distractor set, but simply use the name of the object. In all other cases, we use one of three types of distractor set:

- all the pieces needed to build a target object;
- all the objects referred to since the last mention of this object; or
- all the concrete objects in the world.

The first type of set is used if the object under consideration (OUC) is a negative reference to a piece in context of the creation of a target object. In all other cases, the second type is used if the OUC has been mentioned before and the third type if it has not.

When choosing a referring expression, we first process the distractor set, comparing the properties of the OUC with the properties of all distractors. If a distractor has a different type from the OUC, it is removed from the distractor set. With all other properties, if the distractor has a different value from the OUC, it is removed from the distractor set, and the OUC's property value is added to the list of properties to use.

We then choose the type of referring expression. We first look for a previous reference (PR) to the OUC, and if one exists, determine whether it was in focus. Depending on the case, we use one of the following reference strategies.

**No PR** If the OUC does not yet exist or we are making a negative reference, we use an indefinite article. If the robot is holding the OUC, we use a deictic reference. If the OUC does exist and there are no distractors, we use a definite; if there are distractors we use an indefinite.

**PR was focal** If the PR was within the same turn, we choose a pronoun for our next reference. If it was in focus but in a previous turn, if

the robot is holding the OUC we use a deictic reference, and if the robot is not holding it, we use a pronoun.

**PR was not focal** If the robot is holding the OUC, we make a deictic reference. Otherwise, if the PR was a pronoun, definite, or deictic, we use a definite article. If the PR was indefinite and there are no distractors, we use a definite article, if there are distractors, we use an indefinite article.

If there are any properties in the list, and the reference chosen is not a pronoun, we add them.

### 3.2 Examples of the Reference Algorithm

We will illustrate the reference-selection strategy with two cases from the dialogue in Figure 2.

#### Utterance 4 “a yellow cube”

This object is going to be referred to in a negative context as part of a windmill under construction, so the distractor set is the set of objects needed to make a windmill: {red cube, blue cube, small slat, small slat, green bolt, red bolt}.

We select the properties to use in describing the object under consideration, processing the distractor set. We first remove all objects which do not share the same type as our object under consideration, which leaves {red cube, blue cube}. We then compare the other attributes of our new object with the remaining distractors - in this case “colour”. Since neither cube shares the colour “yellow” with the target object, both are removed from the distractor set, and “yellow” is added to the list of properties to use.

There is no previous reference to this object, and since we are making a negative reference, we automatically choose an indefinite article. We therefore select the reference “a yellow cube”.

#### Utterance 6 “it” (a green bolt)

This object has been referred to before, earlier in the same utterance, so the distractor set is all the references between the earlier one and this one—{red cube}. Since this object has a different type from the bolt we want to describe, the distractor set is now empty, and nothing is added to the list of properties to use.

There is a previous definite reference to the object in the same utterance: “the green bolt”. This reference was focal, so we are free to use a pronoun if appropriate. Since the previous reference

was definite, and the object being referred to does exist, we choose to use a pronoun. We therefore select the reference “it”.

## 4 Experiment Design

In the context of the HRI system, a constant reference strategy is sufficient in that it makes it possible for the robot’s partner to know which item is needed. On the other hand, while the varied forms produced by the more complex mechanism can increase the naturalness of the system output, they may actually be insufficient if they are not used in appropriate current circumstances—for example, “this cube” is not a particularly helpful reference if a user has no way to tell which “this” is. As a consequence, the system for generating such references must be sensitive to the current state of joint actions and—in effect—of joint attention. The difference between the two systems is a test of the adaptive version’s ability to adjust expressions to pertinent circumstances. It is known that people respond well to reduced expressions like “this cube” or “it” when another person uses them appropriately (Bard et al., 2008); we need to see if the robot system can also achieve the benefits that situated reference could provide.

To address this question, the human-robot dialogue system was evaluated through a user study in which subjects interacted with the complete system. Using a between-subjects design, this study compared the two reference strategies, measuring the users’ subjective reactions to the system along with their overall performance in the interaction. Based on the findings from the user evaluation described in (Foster et al., 2009a)—in which the primary effect of varying the reference strategy was on the users’ subjective opinion of the robot—the main prediction for this study was as follows:

- Subjects who interact with a system using adaptive references will rate the quality of the robot’s conversation more highly than the subjects who hear constant references.

We made no specific prediction regarding the effect of reference strategy on any of the objective measures: based on the results of the user evaluation mentioned above, there is no reason to expect an effect either way. Note that—as mentioned above—if the adaptive version makes incorrect choices, that may have a negative impact on users’ ability to understand the system’s generated references. For this reason, even a finding of

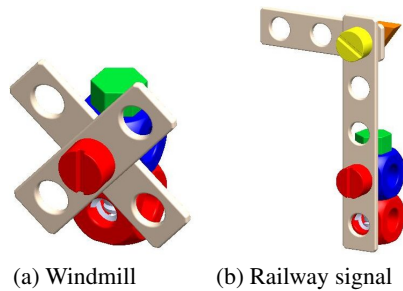


Figure 3: Target objects for the experiment

no objective difference would demonstrate that the adaptive references did not harm the users' ability to interact with the system, as long as it was accompanied by the predicted improvement in subjective judgements.

#### 4.1 Subjects

41 subjects (33 male) took part in this experiment. The mean age of the subjects was 24.5, with a minimum of 19 and a maximum of 42. Of the subjects who indicated an area of study, the two most common areas were Mathematics (14 subjects) and Informatics (also 14 subjects). On a scale of 1 to 5, subjects gave a mean assessment of their knowledge of computers at 4.1, of speech-recognition systems at 2.0, and of human-robot systems at 1.7. Subjects were compensated for their participation in the experiment.

#### 4.2 Scenario

This study used a between-subjects design with one independent variable: each subject interacted either with a system that used a constant strategy to generate referring expressions (19 subjects), or else with a system that used an adaptive strategy (22 subjects).<sup>1</sup>

Each subject built two objects in collaboration with the system, always in the same order. The first target object was the windmill (Figure 3a); after the windmill was completed, the robot and human then built a railway signal (Figure 3b). For both target objects, the user was given a building plan (on paper). To induce an error, both of the plans given to the subjects instructed them to use an incorrect piece: a yellow cube instead of a red cube for the windmill, and a long (seven-hole) slat instead of a medium (five-hole) slat for the rail-

<sup>1</sup>The results of an additional three subjects in the constant-reference condition could not be analysed due to technical difficulties.

way signal. The subjects were told that the plan contained an error and that the robot would correct them when necessary, but did not know the nature of the error.

When the human picked up or requested an incorrect piece during the interaction, the system detected the error and explained to the human what to do in order to assemble the target object correctly. When the robot explained the error and when it handed over the pieces, it used referring expressions that were generated using the constant strategy for half of the subjects, and the adaptive strategy for the other half of the subjects.

#### 4.3 Experimental Set-up and Procedure

The participants stood in front of the table facing the robot, equipped with a headset microphone for speech recognition. The pieces required for the target object—plus a set of additional pieces in order to make the reference task more complex—were placed on the table, using the same layout for every participant. The layout was chosen to ensure that there would be points in the interaction where the subjects had to ask the robot for building pieces from the robot's workspace, as well as situations in which the robot automatically handed over the pieces. Along with the building plan mentioned above, the subjects were given a table with the names of the pieces they could build the objects with.

#### 4.4 Data Acquisition

At the end of a trial, the subject responded to a usability questionnaire consisting of 39 items, which fell into four main categories: Intelligence of the robot (13 items), Task ease and task success (12 items), Feelings of the user (8 items), and Conversation quality (6 items). The items on the questionnaire were based on those used in the user evaluation described in (Foster et al., 2009b), but were adapted for the scenario and research questions of the current study. The questionnaire was presented using software that let the subjects choose values between 1 and 100 with a slider. In addition to the questionnaire, the trials were also video-taped, and the system log files from all trials were kept for further analysis.

### 5 Results

We analysed the data resulting from this study in three different ways. First, the subjects' responses

Table 1: Overall usability results

	Constant	Adaptive	M-W
Intell.	79.0 (15.6)	74.9 (12.7)	$p = 0.19$ , n.s.
Task	72.7 (10.4)	71.1 (8.3)	$p = 0.69$ , n.s.
Feeling	66.9 (15.9)	66.8 (14.2)	$p = 0.51$ , n.s.
Conv.	66.1 (13.6)	75.2 (10.7)	$p = 0.036$ , sig.
Overall	72.1 (11.2)	71.8 (9.1)	$p = 0.68$ , n.s.

to the questionnaire items were compared to determine if there was a difference between the responses given by the two groups. A range of summary objective measures were also gathered from the log files and videos—these included the duration of the interaction measured both in seconds and in system turns, the subjects’ success at building each of the target objects, the number of times that the robot had to explain the construction plan to the user, and the number of times that the users asked the system to repeat its instructions. Finally, we compared the results on the subjective and objective measures to determine which of the objective factors had the largest influence on subjective user satisfaction.

### 5.1 Subjective Measures

The subjects in this study gave a generally positive assessment of their interactions with the system on the questionnaire—with a mean overall satisfaction score of 72.0 out of 100—and rated the perceived intelligence of the robot particularly highly (overall mean of 76.8). Table 1 shows the mean results from the two groups of subjects for each category on the user-satisfaction questionnaire, in all cases on a scale from 0–100 (with the scores for negatively-posed questions inverted).

To test the effect of reference strategy on the usability-questionnaire responses, we performed a Mann-Whitney test comparing the distribution of responses from the two groups of subjects on the overall results, as well as on each sub-category of questions. For most categories, there was no significant difference between the responses of the two groups, with  $p$  values ranging from 0.19 to 0.69 (as shown in Table 1). The only category where a significant difference was found was on the questionnaire items that asked the subjects to assess the robot’s quality as a conversational partner; for those items, the mean score from subjects who heard the adaptive references was significantly higher ( $p < 0.05$ ) than the mean score from the subjects who heard references generated by the constant reference module. Of the six ques-

Table 2: Objective results (all differences n.s.)

Measure	Constant	Adaptive	M-W
Duration (s.)	404.3 (62.8)	410.5 (94.6)	$p = 0.90$
Duration (turns)	29.8 (5.02)	31.2 (5.57)	$p = 0.44$
Rep requests	0.26 (0.45)	0.32 (0.78)	$p = 0.68$
Explanations	2.21 (0.63)	2.41 (0.80)	$p = 0.44$
Successful trials	1.58 (0.61)	1.55 (0.74)	$p = 0.93$

tions that were related to the conversation quality, the most significant impact was on the two questions which assessed the subjects’ understanding of what they were able to do at various points during the interaction.

### 5.2 Objective Measures

Based on the log files and video recordings, we computed a range of objective measures. These measures were divided into three classes, based on those used in the PARADISE dialogue-system evaluation framework (Walker et al., 2000):

- Two **dialogue efficiency** measures: the mean duration of the interaction as measured both in seconds and in system turns;
- Two **dialogue quality** measures: the number of times that the robot gave explanations, and the number of times that the user asked for instructions to be repeated; and
- One **task success** measure: how many of the (two) target objects were constructed as intended (i.e., as shown in Figure 3).

For each of these measures, we tested whether the difference in reference strategy had a significant effect, again via a Mann-Whitney test. Table 2 illustrates the results on these objective measures, divided by the reference strategy.

The results from the two groups of subjects were very similar on all of these measures: on average, the experiment took 404 seconds (nearly seven minutes) to complete with the constant strategy and 410 seconds with the adaptive, the mean number of system turns was close to 30 in both cases, just over one-quarter of all subjects asked for instructions to be repeated, the robot gave just over two explanations per trial, and about three-quarters of all target objects (i.e. 1.5 out of 2) were correctly built. The Mann-Whitney test confirms that none of the differences between the two groups even came close to significance on any of the objective measures.



### 5.3 Comparing Objective and Subjective Measures

In the preceding sections, we presented results on a number of objective and subjective measures. While the subjects generally rated their experience of using the system positively, there was some degree of variation, most of which could not be attributed to the difference in reference strategy. Also, the results on the objective measures varied widely across the subjects, but again were not generally affected by the reference strategy. In this section, we examine the relationship between these two classes of measures in order to determine which of the objective measures had the largest effect on users' subjective reactions to the HRI system.

Being able to predict subjective user satisfaction from more easily-measured objective properties can be very useful for developers of interactive systems: in addition to making it possible to evaluate systems based on automatically available data without the need for extensive experiments with users, such a performance function can also be used in an online, incremental manner to adapt system behaviour to avoid entering a state that is likely to reduce user satisfaction (Litman and Pan, 2002), or can be used as a reward function in a reinforcement-learning scenario (Walker, 2000).

We employed the procedure used in the PARADISE evaluation framework (Walker et al., 2000) to explore the relationship between the subjective and objective factors. The PARADISE model uses stepwise multiple linear regression to predict subjective user satisfaction based on measures representing the performance dimensions of task success, dialogue quality, and dialogue efficiency, resulting in a predictor function of the following form:

$$Satisfaction = \sum_{i=1}^n w_i * \mathcal{N}(m_i)$$

The  $m_i$  terms represent the value of each measure, while the  $\mathcal{N}$  function transforms each measure into a normal distribution using  $z$ -score normalisation. Stepwise linear regression produces coefficients ( $w_i$ ) describing the relative contribution of each predictor to the user satisfaction. If a predictor does not contribute significantly, its  $w_i$  value is zero after the stepwise process.

Table 3 shows the predictor functions that were derived for each of the classes of subjective mea-

asures in this study, using all of the objective measures from Table 2 as initial factors. The  $R^2$  column indicates the percentage of the variance in the target measure that is explained by the predictor function, while the *Significance* column gives significance values for each term in the function.

In general, the two factors with the biggest influence on user satisfaction were the number of repetition requests (which had a uniformly negative effect on user satisfaction), and the number of target objects correctly built by the user (which generally had a positive effect). Aside from the questions on user feelings, the  $R^2$  values are generally in line with those found in previous PARADISE evaluations of other dialogue systems (Walker et al., 2000; Litman and Pan, 2002), and in fact are much higher than those found in a previous similar study (Foster et al., 2009b).

## 6 Discussion

The subjective responses on the relevant items from the usability questionnaire suggest that the subjects perceived the robot to be a better conversational partner if it used contextually varied, situationally-appropriate referring expressions than if it always used a baseline, constant strategy; this supports the main prediction for this study. The result also agrees with the findings of a previous study (Foster et al., 2009a)—this system did not incorporate goal inference and had a less-sophisticated reference strategy, but the main effect of changing reference strategy was also on the users' subjective opinions of the robot's interactive ability. These studies together support the current effort in the natural-language generation community to devise more sophisticated reference generation algorithms.

On the other hand, there was no significant difference between the two groups on any of the objective measures: the dialogue efficiency, dialogue quality, and task success were nearly identical across the two groups of subjects. A detailed analysis of the subjects' gaze and object-manipulation behaviour immediately after various forms of generated references from the robot also failed to find any significant differences between the various reference types. These overall results are not particularly surprising: studies of human-human dialogue in a similar joint construction task (Bard et al., In prep.) have demonstrated that the collaborators preserve quality of construction in

Table 3: PARADISE predictor functions for each category on the usability questionnaire

Measure	Function	$R^2$	Significance
Intelligence	$76.8 + 7.00 * \mathcal{N}(\text{Correct}) - 5.51 * \mathcal{N}(\text{Repeats})$	0.39	Correct: $p < 0.001$ , Repeats: $p < 0.005$
Task	$72.4 + 3.54 * \mathcal{N}(\text{Correct}) - 3.45 * \mathcal{N}(\text{Repeats}) - 2.17 * \mathcal{N}(\text{Explain})$	0.43	Correct: $p < 0.005$ , Repeats: $p < 0.01$ , Explain: $p \approx 0.10$
Feeling	$66.9 - 6.54 * \mathcal{N}(\text{Repeats}) + 4.28 * \mathcal{N}(\text{Seconds})$	0.09	Repeats: $p < 0.05$ , Seconds: $p \approx 0.12$
Conversation	$71.0 + 5.28 * \mathcal{N}(\text{Correct}) - 3.08 * \mathcal{N}(\text{Repeats})$	0.20	Correct: $p < 0.01$ , Repeats: $p \approx 0.10$
Overall	$72.0 + 4.80 * \mathcal{N}(\text{Correct}) - 4.27 * \mathcal{N}(\text{Repeats})$	0.40	Correct: $p < 0.001$ , Repeats: $p < 0.005$

all cases, though circumstances may dictate what strategies they use to do this. Combined with the subjective findings, this lack of an objective effect suggests that the references generated by the adaptive strategy were both sufficient and more natural than those generated by the constant strategy.

The analysis of the relationship between the subjective and objective measures analysis has also confirmed and extended the findings from a similar analysis (Foster et al., 2009b). In that study, the main contributors to user satisfaction were user repetition requests (negative), task success, and dialogue length (both positive). In the current study, the primary factors were similar, although dialogue length was less prominent as a factor and task success was more prominent. These findings are generally intuitive: subjects who are able to complete the joint construction task are clearly having more successful interactions than those who are not able to complete the task, while subjects who need to ask for instructions to be repeated are equally clearly not having successful interactions. The findings add evidence that, in this sort of task-based, embodied dialogue system, users enjoy the experience more when they are able to complete the task successfully and are able to understand the spoken contributions of their partner, and also suggest that designers should concentrate on these aspects of the interaction when designing the system.

## 7 Conclusions

We have presented the reference generation module of a hybrid human-robot interaction system that combines a goal-inference component based on sub-symbolic dynamic neural fields with a natural-language interface based on more traditional symbolic techniques. This combination of approaches results in a system that is able to work

together with a human partner on a mutual construction task, interpreting its partner's verbal and non-verbal behaviour and responding appropriately to unexpected actions (errors) of the partner.

We have then described a user evaluation of this system, concentrating on the impact of different techniques for generating situated references in the context of the robot's corrective feedback. The results of this study indicate that using an adaptive strategy to generate the references significantly increases the users' opinion of the robot as a conversational partner, without having any effect on any of the other measures. This result agrees with the findings of the system evaluation described in (Foster et al., 2009a), and adds evidence that sophisticated generation techniques are able to improve users' experiences with interactive systems.

An analysis of the relationship between the objective and subjective measures found that the main contributors to user satisfaction were the users' task performance (which had a positive effect on most measures of satisfaction), and the number of times the users had to ask for instructions to be repeated (which had a generally negative effect). Again, these results agree with the findings of a previous study (Foster et al., 2009b), and also suggest priorities for designers of this type of task-based interactive system.

## Acknowledgements

This research was supported by the European Commission through the JAST<sup>2</sup> (IST-FP6-003747-IP) and INDIGO<sup>3</sup> (IST-FP6-045388) projects. Thanks to Pawel Dacka and Levent Kent for help in running the experiment and analysing the data.

<sup>2</sup><http://www.jast-project.eu/>

<sup>3</sup><http://www.ics.forth.gr/indigo/>

## References

- E. G. Bard, R. Hill, and M. E. Foster. 2008. What tunes accessibility of referring expressions in task-related dialogue? In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci 2008)*. Chicago.
- E. G. Bard, R. L. Hill, M. E. Foster, and M. Arai. In prep. How do we tune accessibility in joint tasks: Roles and regulations.
- H. Bekkering, E.R.A. de Bruijn, R.H. Cuijpers, R. Newman-Norlund, H.T. van Schie, and R. Meulenbroek. 2009. Joint action: Neurocognitive mechanisms supporting human interaction. *Topics in Cognitive Science*, 1(2):340–352.
- E. Bicho, L. Louro, N. Hipolito, and W. Erlhagen. 2009. A dynamic field approach to goal inference and error monitoring for human-robot interaction. In *Proceedings of the Symposium on “New Frontiers in Human-Robot Interaction”, AISB 2009 Convention*. Heriot-Watt University Edinburgh.
- H. H. Clark. 1996. *Using Language*. Cambridge University Press.
- R. Dale and E. Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- W. Erlhagen and E. Bicho. 2006. The dynamic neural field approach to cognitive robotics. *Journal of Neural Engineering*, 3(3):R36–R54.
- M. E. Foster, E. G. Bard, R. L. Hill, M. Guhe, J. Oberlander, and A. Knoll. 2008a. The roles of haptic-ostensive referring expressions in cooperative, task-based human-robot dialogue. In *Proceedings of HRI 2008*.
- M. E. Foster, M. Giuliani, A. Isard, C. Matheson, J. Oberlander, and A. Knoll. 2009a. Evaluating description and reference strategies in a cooperative human-robot dialogue system. In *Proceedings of IJCAI-09*.
- M. E. Foster, M. Giuliani, and A. Knoll. 2009b. Comparing objective and subjective measures of usability in a human-robot dialogue system. In *Proceedings of ACL-IJCNLP 2009*.
- M. E. Foster, M. Giuliani, T. Müller, M. Rickert, A. Knoll, W. Erlhagen, E. Bicho, N. Hipólito, and L. Louro. 2008b. Combining goal inference and natural-language dialogue for human-robot joint action. In *Proceedings of the 1st International Workshop on Combinations of Intelligent Methods and Applications at ECAI 2008*.
- M. Giuliani and A. Knoll. 2008. MultiML: A general-purpose representation language for multimodal human utterances. In *Proceedings of ICMi 2008*.
- J. D. Kelleher and G.-J. M. Kruijff. 2006. Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of COLING-ACL 2006*.
- A. Kranstedt and I. Wachsmuth. 2005. Incremental generation of multimodal deixis referring to objects. In *Proceedings of ENLG 2005*.
- S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6(3&4):323–340.
- D. J. Litman and S. Pan. 2002. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12(2–3):111–137.
- A. J. N. van Breemen. 2005. iCat: Experimenting with animabotics. In *Proceedings of AISB 2005 Creative Robotics Symposium*.
- I. F. van der Sluis. 2005. *Multimodal Reference: Studies in Automatic Generation of Multimodal Referring Expressions*. Ph.D. thesis, University of Tilburg.
- M. Walker, C. Kamm, and D. Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3–4):363–377.
- M. A. Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.
- M. White. 2006. Efficient realization of coordinate structures in Combinatory Categorical Grammar. *Research on Language and Computation*, 4(1):39–75.



# Towards a Programmable Instrumented Generator

**Chris Mellish**

Computing Science

University of Aberdeen

AB24 3UE, UK

c.mellish@abdn.ac.uk

## Abstract

In this paper, we propose a general way of constructing an NLG system that permits the systematic exploration of the effects of particular system choices on output quality. We call a system developed according to this model a *Programmable Instrumented Generator* (PIG). Although a PIG could be designed and implemented from scratch, it is likely that researchers would also want to create PIGs based on existing systems. We therefore propose an approach to “instrumenting” an NLG system so as to make it PIG-like. To experiment with the idea, we have produced code to support the “instrumenting” of any NLG system written in Java. We report on initial experiments with “instrumenting” two existing systems and attempting to “tune” them to produce text satisfying complex stylistic constraints.

## 1 Introduction

Existing NLG systems are often fairly impenetrable pieces of code. It is hard to see what an NLG system is doing and usually impossible to drive it in any way other than what was originally envisaged. This is particularly unfortunate if the system is supposed to produce text satisfying complex stylistic requirements. Even when an NLG system actually performs very well, it is hard to see why this is or how particular generator decisions produce the overall effects. We propose a way of building systems that will permit more systematic exploration of decisions and their consequences, as well as better exploitation of machine learning to make these decisions better. We call a system built in this way a Programmable Instrumented Genera-

tor (PIG). As an initial exploration of the PIG idea, we have developed a general way of partially instrumenting any NLG system written in Java and have carried out two short experiments with existing NLG systems.

## 2 Controlling an NLG System: Examples

NLG systems are frequently required to produce output that conforms to particular stylistic guidelines. Often conformance can only be tested at the end of the NLG pipeline, when a whole number of complex strategic and tactical decisions have been made, resulting in a complete text. A number of recent pieces of work have begun to address the question of how to tune systems in order to make the decisions that lead to the most stylistically preferred outputs.

Paiva and Evans (2005) (henceforth PE) investigate controlling generator decisions for achieving stylistic goals, e.g. choices between:

The patient takes the two gram dose of the patient’s medicine twice a day.

and

The dose of the patient’s medicine is taken twice a day. It is two grams.

In this case, a stylistic goal of the system is expressed as goal values for features  $SS_i$ , where each  $SS_i$  expresses something that can be measured in the output text, e.g. counting the number of pronouns or passives. The system learns to control the number of times specific binary generator deci-

sions are made ( $GD_j$ ), where these decisions involve things like whether to split the input into 2

than any other connective) and NEGATION (negate a verb and replace it by its antonym). For

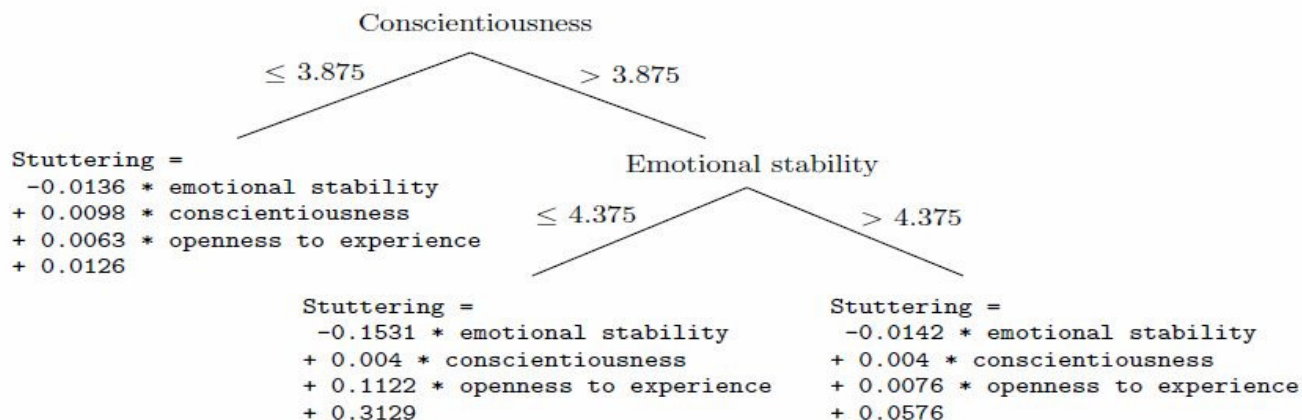


Figure 1: Example PERSONAGE rule

sentences or whether to generate an N PP clause. A process of *offline training* is first used to establish correspondences between counts of generator decisions and the values of the stylistic features. This works by running the system with multiple outputs (making decisions in many possible ways) and keeping track of both the counts of the decisions and also the values of the stylistic features achieved. From this data the system then learns correlations between these:

$$SS_i \cong SS_i^{est} = x_0 + \sum_j x_j \cdot GD_j$$

To actually generate a text given stylistic goals  $SS_i$ , the system then uses an *online control* regime. At each choice point, it considers making  $GD_j$  versus not  $GD_j$ . For each of these two, it estimates all the  $SS_i$  that will be obtained for the complete text, using the learned equations. It prefers the choice that minimises the sum of absolute differences between these and the goal  $SS_i$ , but is prepared to backtrack if necessary (best-first search).

Mairesse and Walker (2008) (henceforth MW) use a different method for tuning their NLG system (“PERSONAGE”), whose objective is to produce texts in the styles of writers with different personality types. In this case, the system performance depends on 67 parameters, e.g. REPETITIONS (whether to repeat existing propositions), PERIOD (leave two sentences connected just with “.”, rather

MW, *offline training* involves having the program generate a set of outputs with random values for all the parameters. Human judges estimate values for the “big five” personality traits (e.g. extroversion, neuroticism) for each output. Machine learning is then used to generate rules to predict how the parameter values depend on the big five numbers. For instance, Figure 1 shows the rule predicting the STUTTERING parameter.

Once these rules are learned, *online control* to produce text according to a given personality (specified by numerical values for the big five traits) uses the learned models to set the parameters, which then determine NLG system behaviour. Human judges indeed recognise these personalities in the texts.

### 3 Towards a PIG

Looking at the previous two examples, one can detect some common features which could be used in other situations:

- An NLG system able to generate random (or all possible) outputs
- Outputs which can be evaluated (by human or machine)
- The logging of key NLG parameters/choices
- Learning of connections between parameters and output quality

This then being used to drive the system to achieve specific goals more efficiently than before.

PE and MW both constructed special NLG systems for their work. One reason for this was that both wanted to ensure that the underlying NLG system allowed the kinds of stylistic variation that would be relevant for their applications. But also, in order to be able to track the choices made by a generator, Paiva and Evans had to implement a new system that kept an explicit record of choices made. This new system also had to be able to organise the search through choices according to a best-first search (it was possibly the first NLG system to be driven in this way). The only possibility for them was to implement a new special purpose generator for their domain with the desired control characteristics.

NLG systems are not usually immediately suitable for tuning of this kind because they make choices that are not exposed for external inspection. Also the way in which choices are made and the overall search strategy is usually hardwired in a way that prevents easy changing. It seems plausible that the approaches of PE and MW would work to some extent for *any* NLG system that can tell you about its choices/ parameter settings, and for *any* stylistic goal whose success can be measured in the text. Moreover, these two are not the only ways one might train/guide an NLG system from such information (for instance, Hovy’s (1990) notion of “monitoring” would be an alternative way of using learned rules to drive the choices of an NLG system). It would be revealing if one could easily compare different control regimes in a single application (e.g. monitoring for PE’s task or best-first search for MW’s), but this is currently difficult because the different systems already have particular control built in.

This discussion motivates the idea of developing a general methodology for the development of NLG systems that permits the systematic exploration of learning and control possibilities. We call a system built in such a way a *Programmable Instrumented Generator* (PIG).<sup>1</sup> A PIG would be an NLG sys-

<sup>1</sup> If one had a sufficiently expressive PIG then perhaps one could train it for *any* testable stylistic goals – a kind of “universal” NLG system?

tem that implements standard NLG algorithms and competences but which is organised in a way that permits inspection and reuse. It would be *instrumented*, in that one would be able to track the choices made in generating a text or texts, in order to tune the performance. It would also be *programmable* in that it would be possible to drive the system in different ways according to a learned (or otherwise determined) “policy”, e.g. to:

- Generate all solutions (overgeneration)
- Generate solutions with some choices fixed/constrained
- Generate solutions with user control of some decisions
- Generate solutions using an in-built choice mechanism
- Generate solutions according to some global search strategy (e.g. monitoring, best-first search)

#### 4 Using a PIG

A general way of using a PIG is shown in Figure 2. A PIG interacts with a (conceptually) separate processing component, which we call the “oracle”. This applies a *policy* to make choices for the generator and receives evaluations of generated texts. It logs the choices made and (using machine learning) can use this information to influence the policy.

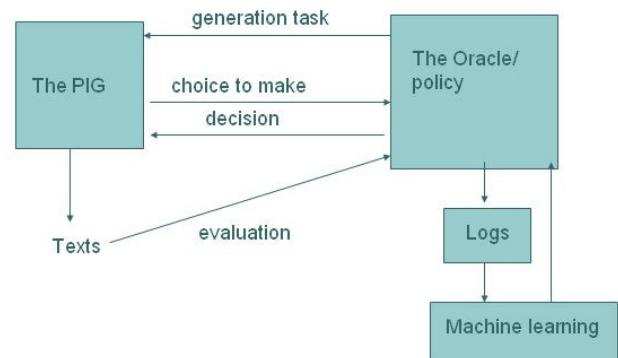


Figure 2: Using a PIG

There are two main modes in which the PIG can be run, though mixtures are also possible. In (offline) *training* mode, the system is run on multiple inputs and uses random or exhaustive search to sample the space of generatable texts. The choices made

are logged, as is the quality of the outputs generated. In (online) *execution* mode, the PIG is run as a normal generator, running on a single input and making choices according to a learned policy.

To support this, the PIG itself needs minimally to support provide external access to the following function:

```
generate(input:InputSpec) returns text:String
```

which produces a text, from a given input specification. On the other hand, the Oracle needs to provide external access to at least the following (used by the PIG):

```
choice(question:String, suggestion:int,
        possibilities:ListOfString, state:String)
returns decision:int or RESTART
```

```
outcome(state:String, value:Float) (no return value)
```

where *question* represents a choice to be made (with possible answers *possibilities*), *suggestion* is the index of a suggested choice and *decision* is the index of the choice made. *state* is a representation of generator state, in some standard format (e.g. ARFF (Hall et al 2009)) and *outcome* (giving the final state and the text quality) is called as the last action of generating a text. *RESTART* is a special value that by convention causes the system to return to a state where it can be asked to generate another text.

To support the above, the PIG needs to maintain some representation of program state. Also the oracle needs to implement a training/testing algorithm that involves providing the PIG with example inputs, restarting the PIG on the current or a new example, implementing a policy, logging results and possibly interacting with a user.

The above model of how to use a PIG is partly motivated by existing approaches to monitoring and testing complex electronic equipment. Testing is often carried out by attaching “automatic test equipment” to the unit under test. This automatic test equipment is akin to our “oracle” in that it drives the unit through special test sequences and automatically records what is going on.

## 5 The PIG panel

There is a practical question of how best to build PIGs and what resources there might be to support this. Given their concern with explicit representation of choices, NLG models based on Systemic Grammar (Bateman 1997) might well be promising as a general framework here. But in reality, NLG systems are built using many different theoretical approaches, and most decisions are hard-coded in a conventional programming language. In order to investigate the PIG concept further, therefore, we have developed a general way of “instrumenting” in a limited way any NLG system written in Java (giving rise to a *PIGlet*). We have also implemented a general enough oracle for some initial experiments to be made with a couple of PIGlets. This experimental work is in line with the API given above but implemented in a way specific to the Java language.

In order to instrument the client generator, one has to identify places where interesting choices are made. This is obviously best done by someone with knowledge of the system. There are a number of ways to do this, but the simplest basically replaces a construct of the form:

```
if (<condition>) <action>
```

by

```
if (Oracle.condRec(<name>,<condition>)) <action>
```

where *<name>* is a string naming this particular choice. This allows the oracle to intervene when the choice is made, but possibly taking into account the suggested answer (*<condition>*).

The implemented oracle (the “PIG panel”) supports a kind of “single stepping” of the generator (between successive choices), manual control of choices and restarting. It has built in policies which include random generation, following the choices suggested by the PIGlet, systematic generation of all possibilities (depth-first) and SARSA, a kind of reinforcement learning (Sutton and Barto 1998). It provides simple statistics about the evaluations of the texts generated using the current policy and a user interface (Figure 3).





**Figure 3: PIG Panel interface**

For the oracle to be able to control the PIGlet, it needs to be provided with a “connector” which represents it through a standard API (specifying how to generate a text, how to evaluate a text, what examples can be used, etc.). This also includes a specification of how to derive the “state” information about the generator which is logged for machine learning process. State information can include the number of times particular choices are made (as in PE), the most recent choices made and other generator-specific parameters which are communicated to the oracle (as in MW).

Finally the PIGlet and oracle are linked via a “harness” which specifies the basic mode of operation (essentially training vs execution).

In the following sections, we describe two tentative experiments which produced PIGlets from existing NLG systems and investigated the use of the PIG panel to support training of the system. It is important to note that for these systems the instrumenting was done by someone (the author) with limited knowledge of the underlying NLG system and with a notion of text quality different from that used by the original system. Also, in both cases the limited availability of example data meant that testing had to be performed on the training data (and so any positive results may be partly due to overfitting).

## 6 Experiment 1: Matching human texts

For this experiment, we took an NLG system that produces pollen forecasts and was written by Ross Turner (Turner et al 2006). Turner collected 68

examples of pollen prediction data for Scotland (each consisting of 6 small integers and a characterisation of the previous trend) with human-written forecasts, which we took as both our training and test data. We evaluated text quality by similarity to the human text, as measured by the Meteor metric (Lavie and Denkowski 2009). Note that the human forecasters had access to more background knowledge than the system, and so this is not a task that the system would be expected to do particularly well on.

The notion of program “state” that the oracle logged took the form of the 6 input values, together with the values of 7 choices made by the system (relating to the inclusion of trend information, thresholds for the words “high” and “low”, whether to segment the data and whether to include hay fever information).

The system was trained by generating about 10000 random texts (making random decisions for randomly selected examples). For each, the numerical outcome (Meteor score) and state information was recorded. The half of the resulting data with highest outcomes was extracted and used to predict rules for the 7 choices, given the 6 input parameters (we used Weka (Hall et al 2009) with the JRip algorithm). The resulting rules were transcribed into a specific “policy” (Java class) for the oracle.

Applied to the 68 examples, trying random generation for 3 times on each, the system obtained an average Meteor score of 0.265. Following the original system’s suggestions produced an average score of 0.279. Following the learned policy, the system also obtained an average of 0.279. The difference between the learned behaviour and random generation is significant ( $p=0.002$ ) according to a t test.

## 7 Experiment 2: Text length control

A challenging stylistic requirement for NLG is that of producing a text satisfying precise length requirements (Reiter 2000). For this experiment, we took the EleonPlus NLG system developed by Hien Nguyen. This combines the existing Eleon user interface for domain authoring (Bilidas et al 2007) with a new NLG system that incorporates the SimpleNLG realiser (Gatt and Reiter 2009).

The system was used for a simple domain of texts about university buildings. The data used was the authored information about 7 university buildings and associated objects. We evaluated texts using a simple (character) length criterion, where the ideal text was 250 characters, with a steeply increasing penalty for texts longer than this and a slowly increasing penalty for texts that are shorter.

The notion of “state” that was logged took account of the depth of the traversal of the domain data, the maximum number of facts per sentence and an aggregation decision.

Following the previous successful demonstration of reinforcement learning for NLG decisions (Rieser and Lemon 2006), we decided to use the SARSA approach (though without function approximation) for the training. This involves rewarding individual states for their (direct or indirect) influence on outcome quality as the system actually performs. The policy is a mixture of random exploration and the choosing of the currently most promising states, according to the value of a numerical parameter  $\epsilon$ .

Running the system on the 7 examples with 3 random generations for each produced an average text quality of -2514. We tried a SARSA training regime with 3000 random examples at  $\epsilon=0.1$ , followed by 2000 random examples at  $\epsilon=0.001$ . Following this, we looked at performance on the 7 examples with  $\epsilon=0$ . The average text quality was -149. This was exactly the same quality as that achieved by following the original NLG system’s policy. Even though there is a large difference in average quality between random generation and the learned policy, this is, however, not statistically significant ( $p = 0.12$ ) because of the small number of examples and large variation between text qualities.

## 8 Conclusions and Further Work

Each of our initial experiments was carried out by a single person in less than a week of work, (which included some concurrent development of the PIG panel software and some initial exploration of the underlying NLG system). This shows that it is relatively quick (even with limited knowledge of the original NLG system) for someone to instrument

an existing NLG system and to begin to investigate ways of optimizing its performance (perhaps with different goals than it was originally built for). This result is probably more important than the particular results achieved (though it is promising that some are statistically significant).

Further work on the general software could focus on the issue of the visualization of choices. Here it might be interesting to impose a Systemic network description on the interdependencies between choices, even when the underlying system is built with quite a different methodology.

More important, however, is to develop a better understanding of what sorts of behaviour in an NLG system can be exposed to machine learning to optimize the satisfaction of what kinds of stylistic goals. Also we need to develop methodologies for systematically exploring the possibilities, in terms of the characterization of NLG system state and the types of learning that are attempted. It is to be hoped that software of the kind we have developed here will help to make these tasks easier.

Finally, this paper has described the development and use of PIGs mainly from the point of view of making the best of NLG systems rather like what we already have. The separation of logic and control supported by the PIG architecture could change the way we think about NLG systems in the first place. For instance, a PIG could easily be made to overgenerate (in the manner, for instance, of HALOGEN (Langkilde-Geary 2003)), in the confidence that an oracle could later be devised that appropriately weeded out non-productive paths.

## Acknowledgments

This work was supported by EPSRC grant EP/E011764/1. The ideas here have benefited particularly from discussions with Graeme Ritchie and Roger Evans. We also acknowledge the helpful comments of two anonymous reviewers.

## References

- John Bateman. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering* 3(1):15-55.
- Dimitris Bilidas, MariaTheologou and Vangelis Karkaletsis. 2007. Enriching OWL Ontologies with Linguistic and User-Related Annotations: The ELEON System. *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Patra, Greece.
- Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. *Proceedings of ENLG-2009*.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
- Eduard H. Hovy. 1990. Pragmatics and Natural Language Generation. *Artificial Intelligence* 43(2), pp. 153–198.
- Alon Lavie and Michael Denkowski. 2009. The METEOR Metric for Automatic Evaluation of Machine Translation. *Machine Translation*, published online 1st November 2009.
- Irene Langkilde-Geary. 2003. A foundation for general-purpose natural language generation: sentence realization using probabilistic models of language. PhD thesis, University of Southern California, Los Angeles, USA.
- François Mairesse and Marilyn Walker. 2008. Trainable Generation of Big-Five Personality Styles through Data-driven Parameter Estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus.
- Daniel Paiva and Roger Evans. 2005. Empirically-based control of natural language generation. *Proceedings of the 43rd Annual Meeting of the ACL*, pages 58-65.
- Ehud Reiter. 2000. Pipelines and Size Constraints. *Computational Linguistics*. 26:251-259.
- Verena Rieser and Oliver Lemon. 2006. Using Machine Learning to Explore Human Multimodal Clarification Strategies. *Procs of ACL 2006*.
- Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- Ross Turner, Yaji Sripada, Ehud Reiter and Ian Davy. 2006. Generating Spatio-Temporal Descriptions in Pollen Forecasts. *Proceedings of EACL06 Companion Volume*.



# Using Semantic Web Technology to Support NLG

## Case study: OWL finds RAGS

Chris Mellish

Computing Science

University of Aberdeen, Aberdeen AB24 3UE, UK

c.mellish@abdn.ac.uk

### Abstract

The semantic web is a general vision for supporting knowledge-based processing across the WWW and its successors. As such, semantic web technology has potential to support the exchange and processing of complex NLG data. This paper discusses one particular approach to data sharing and exchange that was developed for NLG – the RAGS framework. This was developed independently of the semantic web. RAGS was relatively complex and involved a number of idiosyncratic features. However, we present a rational reconstruction of RAGS in terms of semantic web concepts, which yields a relatively simple approach that can exploit semantic web technology directly. Given that RAGS was motivated by the concerns of the NLG community, it is perhaps remarkable that its aspirations seem to fit so well with semantic web technology.

### 1 Introduction

The semantic web is a vision of a future world wide web where content, rather than being primarily in the form of unanalysed natural language, is *machine accessible* (Antoniou and van Harmelen, 2004). This could bring a number of advantages compared to the present web, in terms, for instance of the precision of web search mechanisms and the extent to which web resources can be brought together automatically for solving complex processing problems.

From the point of view of NLG, the semantic web offers a vision of a situation where resources can be formally described and composed, and where it is possible to live with the variety of different approaches and views of the world which characterise the users of the web. Given the het-

erogeneous nature of NLG, it seems worth considering whether there might be some useful ideas here for NLG.

The foundation of the semantic web is the idea of replacing formatting-oriented languages such as HTML by varieties of XML which can capture the structure of content explicitly. Markup of linguistic resources (text corpora, transcribed dialogues, etc.) via XML is now standard in NLP, but very often each use of XML is unique and hard to reconcile with any other use. The semantic web goes beyond this in proposing a more abstract basic language and allowing explicit representation of what things in it *mean*. For the semantic web, RDF (Klyne and Carroll, 2003), which is built on top of XML, represents a common language for expressing content as a “semantic network” of triples, and ontology languages, such as OWL (McGuinness and van Harmelen, 2004), allow the expression of constraints and principles which partially constrain possible interpretations of the symbols used in the RDF. These ontologies are statements that themselves can be inspected and modified. They can provide the basis for different people to express their assumptions, agreements and disagreements, and to synthesise complex data from multiple sources.

### 2 RAGS

RAGS (“Reference Architecture for Generation Systems”) was an attempt to exploit previous ideas about common features between NLG systems in order to propose a reference architecture that would help researchers to share, modularise and evaluate NLG systems and their components. In practice, the project found that there was less agreement than expected among NLG researchers on the modules of an NLG system or the order of their running. On the other hand, there *was* reasonable agreement (at an abstract level) about the kinds of *data* that an NLG system needs to repre-

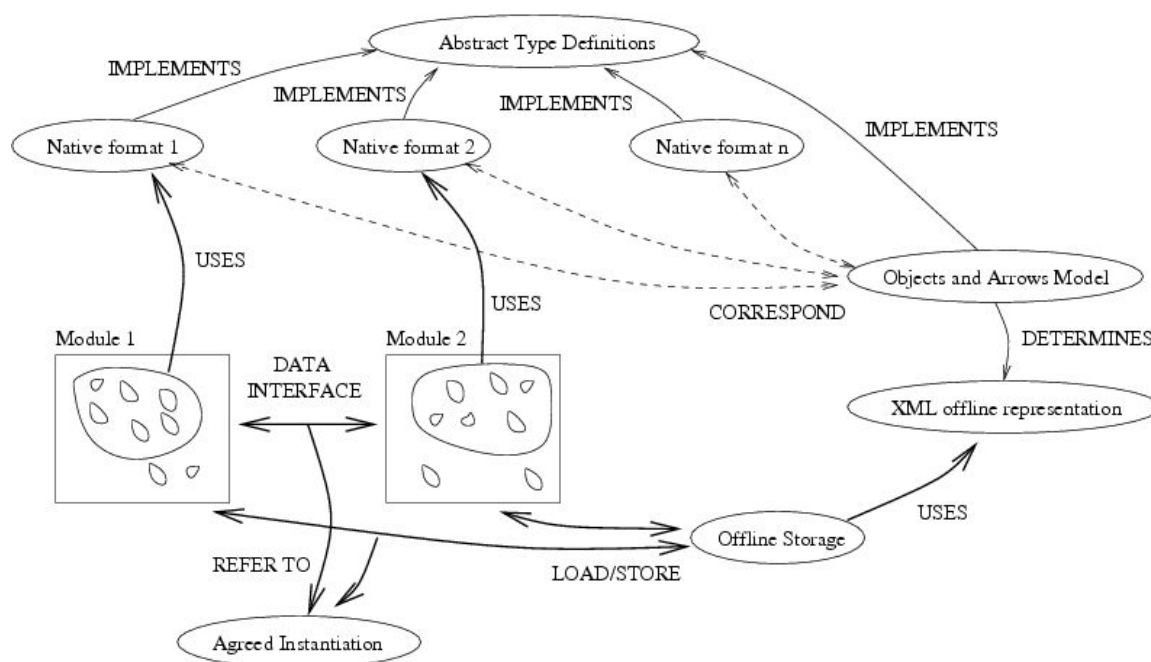


Figure 1: RAGS

sent, in passing from some original non-linguistic input to a fully-formed linguistic description as its output. Figure 1 summarises RAGS and how it was intended to be used. The following description simplifies/ rationalises in a number of ways; more information about RAGS can be found in (Mellish et al., 2006) and (Cahill et al., 2001).

RAGS provides *abstract type definitions* for 6 different types of data representations: conceptual, rhetorical, document, semantic, syntactic and “quote”. As an example, here are the definitions associated with document representations (which describe the parts of a document and aspects of their logical formatting).

$$\begin{aligned}
 \text{DocRep} &= \text{DocAttr} \times \text{DocRepSeq} \\
 \text{DocRepSeq} &= \text{DocRep}^* \\
 \text{DocAttr} &= (\text{DocFeat} \rightarrow \text{DocAtom}) \\
 \text{DocFeat}, \text{DocAtom} &\in \text{Primitives}
 \end{aligned}$$

These type definitions express the types in terms of set theory (using constructions such as union, Cartesian product, subset and function), where the “primitive” types correspond to basic sets that have to be defined in a theory-specific way. Thus a document representation (*DocRep*) has two components, a *DocAttr* and a *DocRepSeq*. A *DocRepSeq* is a sequence of zero or more *DocReps*, which represent the document structure of the parts of the document. A *DocAttr* is a

function from *DocFeats* to *DocAtoms*. The former can be thought of as a set of names of “features” for parts of documents (e.g. text level, indentation) and the latter as a set of values for these (e.g. “Clause”, 3). However the sets *DocFeat* and *DocAtom* are left unspecified in the RAGS formalisation. The idea is that researchers will not necessarily agree how to instantiate these primitives. Clusters of researchers may agree on standard possibilities for these sets and this will help them to share data (but even researchers not able to agree on the primitive sets will be able to understand one another’s data to some extent). When two NLG modules need to exchange data, they need to refer to an agreed instantiation of the primitive types in order to share fully.

Although it gives some examples, RAGS does not specify any particular formats in which data should be represented in different programming languages and used by NLG modules – potentially, arbitrary “native formats” could be used, as long as they can be viewed as “implementations” of the abstract type definitions. Further conditions are, however, imposed by requiring a correspondance between native formats and representations in a “reference implementation” called the *Objects and Arrows* (OA) model. This provides answers to further questions, such as what partially-specified data representations are possi-

ble, where re-entrancy can occur and how data representations of different types can be mixed. The OA model represents data as directed graphs, whose nodes represent typed pieces of data and whose edges represent relations. The possible legal states of an OA representation are formally defined, in a way that resembles the way that information states in a unification grammar can be characterised (Shieber, 1986). Each node in the graph is labelled with a type, e.g. *DocRep*, *DocAtom*. Each node is assumed to have a unique identifier and for primitive types a node can also have a *subtype*, a theory-dependent elaboration that applies to this particular data object (e.g. a *DocAtom* could have the subtype 3). Some edges in the graph indicate “local arrows”, which describe the parts of complex datastructures. For instance, edges labelled *el* indicate elements of unordered structures, and arrows labelled *el-1*, *el-2* etc. indicate components of ordered structures. Edges can also represent “non-local arrows” which describe relationships between representations at different levels. Non-local arrows allow data representations at different levels to be mixed into a single graph.

Representations in the Objects and Arrows model can be mapped to an XML interchange representation. The correspondance between native formats and the OA model can then be used to map between native data representations and XML (in both directions). Modules can communicate via agreed native formats or, if this is undesirable, via the XML representation.

### 3 Some Problems with RAGS

Some of the problems with RAGS, which have impeded its uptake, include:

- Complexity and lack of tools – RAGS was a proposal with a unique shape and takes some time to understand fully. It ploughs its own distinctive furrow. Because it was developed in a project with limited resources, there are limited tools provided for, for instance, displaying RAGS structures, supporting different programming languages and styles and automatic consistency checking. This means that engaging with RAGS involves initially a significant amount of low-level programming, with benefits only to be seen at some time in the future.

- Idiosyncratic use of XML – RAGS had to address the problem of expressing a graph in a serialised form, where there can be multiple, but different, serialisations of the same graph. It did this in its own way, which means that it is hard to exploit general tools which address this problem in other areas.
- Inclarity about how to “buy-in” to limited degrees - there is no defined mechanism for dividing generally agreed from non-agreed elements of a RAGS representation or for expressing or referring to an “agreed instantiation”.

### 4 Recasting RAGS data in terms of RDF

The first step in recasting RAGS in semantic web terms is to exploit the fact that it is the OA model (rather than the abstract type definitions) that is the basis of data communication, since this model expresses more concrete requirements on the form of the data. Therefore initially we concentrate on the OA model and its XML serialisation.

RDF is a data model that fits OA graphs very well. It provides a way of creating “semantic networks” with sets of object-attribute-value triples. Objects and attributes are “resources”, which are associated with Universal Resource Identifiers (URIs), and values are either resources or basic data items (“literals”, e.g. strings or integers). Resources have types, indicated by the RDF `type` attribute. The idea of an RDF resource maps nicely to a RAGS object, and the idea of an RDF attribute maps nicely to a RAGS arrow.

URIs provide a natural way to allow reentrancy to be represented and at the same time permit unambiguous references to external objects in the way that RAGS intended should be possible. The XML namespace mechanism allows complex IDs to be abbreviated by names of the form `Prefix:N`, where `Prefix` is an abbreviation for the place where the name `N` is defined and `N` is the basic name (sometimes the prefix can be inferred from context and can be missed out). Thus, for instance, if the prefix `rags` is defined to stand for the start of a URI identifying RAGS then `rags:DocRep` identifies the type `DocRep` defined by RAGS, as distinct from any other definition anyone might have.

It follows from the preceding discussion that instances of the RAGS abstract types can be mapped naturally to RDF resources with the abstract type

as the value for the RDF attribute `type`. Arrows can be mapped into RDF attributes, and so it really only remains to have a convention for the representation of “subtype” information in RAGS. In this paper, we will assume that instances of primitive types can have a value for the attribute `sub`.

RDF can be serialised in XML in a number of ways (which in fact are closely related to the possible XML serialisations of RAGS).

To summarise, using RDF rather than RAGS XML introduces negligible extra complexity but has a number of advantages:

- Because it is a standard use of XML, it means that generic software tools can be used with it. Existing tools, for instance, support reading and writing RDF from different programming languages, visualising RDF structures (see Figure 4) and consistency checking.
- Because it comes with a universal way of naming concepts, it means that it is possible for different RAGS resources to be unambiguous and reference one another.

## 5 Formalising the RAGS types using ontologies

RDF gives us a more standard way to interpret the OA model and to serialise OA instance information in XML. However, on its own it does not enforce data representations to be consistent with the intent of the abstract type definitions. For instance, it does not prevent an element of a *DocRepSeq* being something other than a *DocRep*.

For RAGS, an XML DTD provided constraints on what could appear in the XML serialisation, but DTDs are not very expressive and the RAGS DTD had to be quite loose in order to allow partial representations. The modern way to define the terms that appear in a use of RDF, and what constraints there are on their use, is to define an *ontology* using a language like RDFS (Brickley and Guha, 2003) or OWL (McGuinness and van Harmelen, 2004). An ontology can be thought of as a set of logical axioms that limits possible interpretations of the terms. This could be used to show, for instance, that a given set of instance data is inconsistent with an ontology, or that further logical consequences follow from it. There are various versions of the web ontology language OWL. In this paper, we use OWL DL, which is based on a description

logic, and we will use standard description logic notation in this paper.

Description logics allow one to make statements about the terms (names of concepts and roles) used in some knowledge representation. In our case, a concept corresponds to a RAGS type (implemented by an RDF resource linked to from individuals by a `type` attribute) and a role corresponds to a RAGS arrow (implemented by an RDF attribute). Complex names of concepts can be built from simple names using particular constructors. For instance, if  $\alpha$  and  $\beta$  are two concept names (simple concept names or more complex expressions made from them) and  $\rho$  is a role name, then, the following are also concept names:

- $\alpha \sqcup \beta$  - names the concept of everything which is  $\alpha$  or  $\beta$
- $\alpha \sqcap \beta$  - names the concept of everything which is  $\alpha$  and  $\beta$
- $\exists \rho. \alpha$  - names the concept of everything which has a value for  $\rho$  which is an instance of concept  $\alpha$
- $\forall \rho. \alpha$  - names the concept of everything which only has values for  $\rho$  which are instances of concept  $\alpha$
- $=_n \rho$  - names the concept of everything with exactly  $n$  different values of  $\rho$

Constructors can be nested, so that, for instance,  $C_1 \sqcap \exists r. C_2$  is a possible concept name, assuming that  $C_1$  and  $C_2$  are simple concept names and  $r$  is a role name.

For an ontology, one then writes logical axioms stating relationships between simple or complex concept names, e.g.

- $\alpha_1 \sqsubseteq \alpha_2$  - states that  $\alpha_1$  names a more specific concept than  $\alpha_2$
- $\alpha_1 \equiv \alpha_2$  - states that  $\alpha_1$  names the same concept as  $\alpha_2$
- $disjoint(\{\alpha_1, \dots, \alpha_n\})$  - states that  $\alpha_1, \dots, \alpha_n$  are disjoint concepts (no pair can have a common instance).
- $\rho_1 \sqsubseteq_r \rho_2$  - states that  $\rho_1$  names a subproperty of  $\rho_2$
- $domain(\rho, \alpha)$  - states that  $\rho$  can only apply to things satisfying concept  $\alpha$
- $range(\rho, \alpha)$  - states that values of  $\rho$  must satisfy concept  $\alpha$
- $functional(\rho)$  - states that  $\rho$  is a functional role



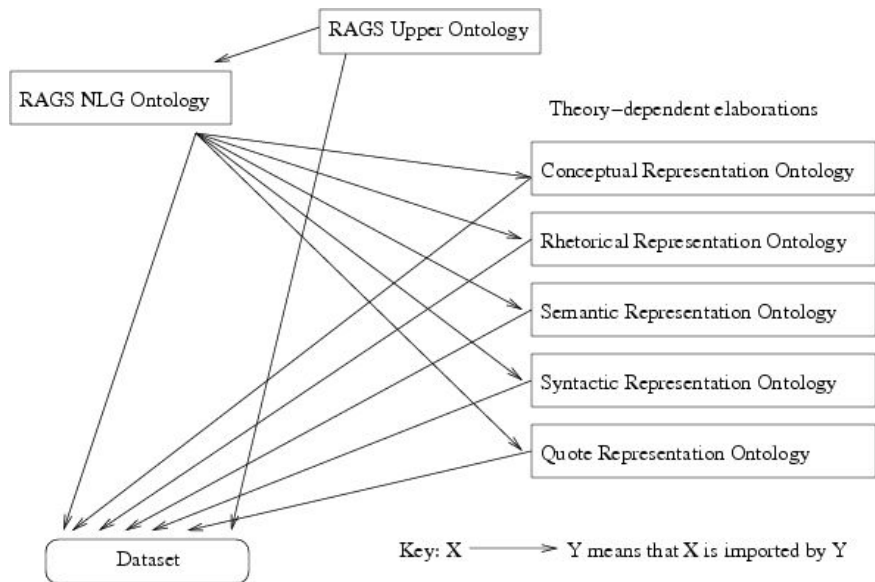


Figure 2: Using Multiple Ontologies in RAGS

For more information on the formal basis of description logics and their relevance for ontologies, see (Horrocks, 2005).

For RAGS, a number of advantages follow from adopting DLs as the basis for formalising data representations:

**Modularity.** A given set of instance data may relate to more than one ontology which expresses constraints on it. One ontology is said to *import* another if it inherits the constraints that the other provides. The standard (monotonic) logic approach applies, in that one can choose to describe the world in terms of any consistent set of axioms. Ontologies package up sets of axioms into bundles that one might decide to include or not include in one’s own model of the world. Ontologies for different purposes can be built by different people but used together in an eclectic way. This formalises the idea of “variable buy-in” in RAGS.

**Openness.** Also corresponding to the usual approach with logic, the semantics of OWL makes no *closed world assumption*. Thus a statement cannot be inconsistent purely by *failing* to specify something. This means that it is only necessary to describe the properties of *complete* datastructures in an ontology. Partial descriptions of data will be not be inconsistent by virtue of their partiality. Only having to describe complete datastruc-

tures makes the specification job much simpler. In a similar way, the semantics of OWL makes no *unique names assumption*. Thus individuals with different names are not necessarily distinct. This means that it is generally possible to make a given description more specific by specifying the identity of two individuals (unless inconsistency arises through, for instance, the individuals having incompatible types). This is another requirement if one wishes the power to add further information to partial representations.

**Software tools.** As with RDF, use of OWL DL opens up the possibility of exploiting generic tools developed elsewhere, for instance reasoners and facilities to translate RAGS concepts into programming language structures.

## 6 The RAGS Ontologies

It is convenient to modularise what RAGS requires as well-formedness constraints as a *set of ontologies*. This allows us to formalise what it means to “buy-in” to one or more parts of RAGS. It simply means importing one or more of the RAGS ontologies (in addition to one’s own) and making use of some of the terms defined in them. We now outline one possible version of the core RAGS ontologies.

Figure 2 shows the way that the RAGS ontologies are intended to be used. A dataset in general makes use of concepts defined in the core RAGS ontologies (the “upper ontology” and the “NLG



Figure 3: The RAGS “NLG ontology”

ontology”<sup>1</sup> and also theory-dependent elaborations defined in separate ontologies (which may correspond one-to-one to the different levels, as shown, but need not do so necessarily). These elaborations are not (initially) provided by RAGS but may arise from arbitrary research subcommunities. Logically, the dataset is simple described/constrained by the union of the axioms coming from the ontologies it makes use of. In general, different datasets will make consistent references to the concepts in the core RAGS ontologies, but they may make use of different theory-dependent elaborations.

The basis of RAGS is a very neutral theory about datatypes (and how they can be encoded in XML). This is in fact independent of the fact that RAGS is intended for NLG - at this level, RAGS could be used to describe data in other domains, or NLG-oriented data that is not covered by RAGS. It therefore makes sense to think of this as a separable part of the theory, the “upper ontology”. At the top level, datastructures (instances of *Object*) belong to one of the concepts *Ordered*, *Set* and *Primitive*. Ordered structures are divided

up in terms of the number of components (concepts *Arity-1*, *Arity-2* etc) and whether they are *Tuples* or *Sequences*. For convenience, union types such as *Arity-atleast-2* are also defined.

The RAGS NLG ontology (see Figure 3 for an overview) contains the main substance of the RAGS type definitions. As the figure shows, it introduces a number of new concepts as subconcepts of the upper ontology concepts. For instance, *DocRepSeq*, *RhetRepSeq*, *Adj* and *Scoping* are introduced as subconcepts of *SpecificSequence* (these concepts correspond to types used in RAGS at the document, rhetorical, syntactic and semantic levels). Not shown in the diagram is the type of roles, *Functional* that includes all arguments of RAGS functional objects<sup>2</sup>. The set of type definitions describing a level of representation in RAGS translates quite directly into a set of axioms in this ontology. For instance, the following is the encoding of the type definitions for document rep-

<sup>1</sup>These are both available in full from <http://www.abdn.ac.uk/~csc248/ontologies/>

<sup>2</sup>Whereas in RAGS a functional type (e.g. *DocAttr*) is represented as an unordered set of (ordered) pairs of the form  $\langle \text{function argument}, \text{function value} \rangle$ , here we can simply implement the function arguments as RDF attributes and omit the functional types.

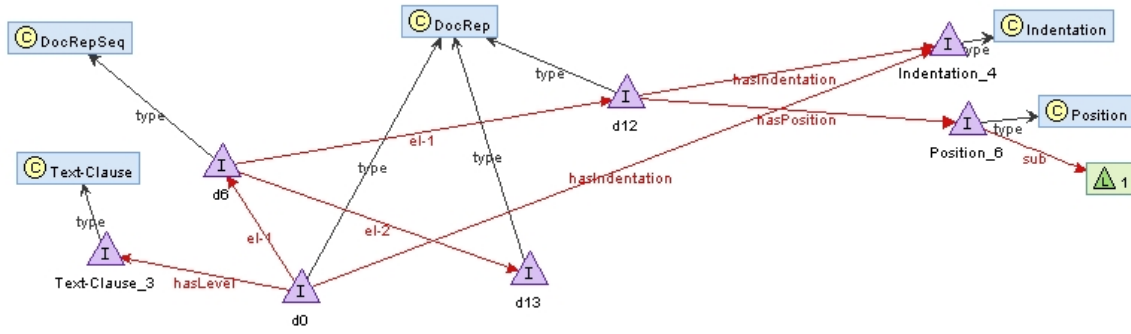


Figure 4: Visualisation of example Document Representation

representations. First of all, it is necessary to specify that a *DocRep* is a tuple with arity 1 (the *DocAttr* is not needed), and that its component must have a specific type:

$$\begin{aligned} \text{DocRep} &\sqsubseteq \text{Tuple} \sqcap \text{Arity-1} \\ \text{DocRep} &\sqsubseteq (\forall \text{el-1. DocRepSeq}) \end{aligned}$$

The next few axioms do a similar job for *DocRepSeq*, a kind of sequence:

$$\begin{aligned} \text{DocRepSeq} &\sqsubseteq \text{SpecificSequence} \\ \text{DocRepSeq} &\sqsubseteq (\forall n\text{-el. DocRep}) \end{aligned}$$

Finally, a high level role *DocFeat* is introduced, whose subroles will correspond to particular document features like *Indentation*. The domain and range of such roles are constrained via constraints on *DocFeat*:

$$\begin{aligned} \text{DocFeat} &\sqsubseteq_r \text{Functional} \\ \text{domain}(\text{DocFeat}, \text{DocRep}) & \\ \text{range}(\text{DocFeat}, \text{DocAtom}) & \end{aligned}$$

## 7 Other Ontologies and RAGS

As stated above, in general one produces specialisations of the RAGS framework by creating new ontologies that:

- Introduce specialisations of the RAGS primitive concepts (and perhaps new roles that instances of these can have).
- Introduce subroles of the RAGS functional roles.
- Add new axioms that specialise existing RAGS requirements, involving the core concepts and roles and/or the newly introduced ones.

An example of this might be an example ontology that instantiates a simple theory of document structure, following (Power et al., 2003). Given the notion of document structure introduced in section 2 and formalised in section 6, it is really only necessary to specify the “features” of pieces of document structure (*DocFeat*) and their “values” (*DocAtom*). The former are modelled as roles and the latter in terms of concepts. First we introduce the basic types of values:

$$\begin{aligned} \text{DocAtom} &\equiv (\text{Position} \sqcup \text{Indentation} \sqcup \\ &\quad \text{Level} \sqcup \text{Connective}) \\ &\text{disjoint}(\{\text{Position}, \text{Indentation}, \text{Level}, \\ &\quad \text{Connective}\}) \end{aligned}$$

Positions in the text could be modelled by objects whose *sub* values are positive integers (there is a standard (RDFS) datatype for these). The following axioms capture this and the characteristics of the role *hasPosition*:

$$\begin{aligned} \text{Position} &\sqsubseteq (\forall \text{sub.xsd} : \text{positiveInteger}) \\ \text{hasPosition} &\sqsubseteq_r \text{DocFeat} \\ \text{range}(\text{hasPosition}, \text{Position}) & \\ \text{functional}(\text{hasPosition}) & \end{aligned}$$

For text levels, on the other hand, there is a fixed set of possible values. These are introduced as disjoint concepts. In addition, the role *hasLevel* is introduced:

$$\begin{aligned} \text{Level} &\equiv (\text{Chapter} \sqcup \text{Paragraph} \sqcup \text{Section} \sqcup \\ &\quad \text{Text-Clause} \sqcup \text{Text-Phrase} \sqcup \\ &\quad \text{Text-Sentence}) \\ &\text{disjoint}(\{\text{Chapter}, \text{Paragraph}, \text{Section}, \\ &\quad \text{Text-Clause}, \text{Text-Phrase}, \text{Text-Sentence}\}) \\ \text{hasLevel} &\sqsubseteq_r \text{DocFeat} \\ \text{range}(\text{hasLevel}, \text{Level}) & \\ \text{functional}(\text{hasLevel}) & \end{aligned}$$

Figure 4 shows an example *DocRep* (labelled “d12”) described by this ontology, as visualised by the RDF-Gravity tool developed by Salzburg Research. It consists of a *DocRepSeq* (“d6”) with

two *DocRep* components (“d0” and “d13”). The indentations of “d12” and “d0” are not known, but they are constrained to be the same.

It is easy to think of examples of other (existing or potential) ontologies that could provide theories of the RAGS primitive types. For instance, WordNet (Miller, 1995) or the Generalised Upper Model (Bateman et al., 1995) could be used to bring in a theory of semantic predicates (*SemPred*). An ontology of rhetorical relations (*RhetRel*) could be built based on RST, and so on.

Ontologies can use the expressive power of OWL to make relatively complex statements. For instance, the following could be used in an RST ontology to capture the concept of nucleus-satellite relations and the constraint that a rhetorical representation with such a relation (as its first component) has exactly two subspans (recorded in the second component):

$$NS \sqsubseteq RhetRel \\ (RhetRep \sqcap \exists el-1.NS) \sqsubseteq (\exists el-2.Arity-2)$$

## 8 Relation to Other Work

Reworking RAGS to use semantic web technology relates to two main strands of previous work: work on XML-based markup of linguistic resources and work on linguistic ontologies.

The trouble with applying existing annotation methods (e.g. the Text Encoding Initiative) to NLG is that they presuppose the existence of a linear text to start with, whereas in NLG one is forced to represent more abstract structures before coming up with the actual text. A recent proposal from Linguistics for a linguistic ontology for the semantic web (Farrar and Langendoen, 2003) is again based around making annotations to existing text. Research is only just beginning to escape from a “time-based” mode of annotation, for instance by using “stand-off” annotations to indicate layout (Bateman et al., 2002). In addition, most annotation schemes are partial (only describe certain aspects of the text) and non-structured (assign simple labels to portions of text). For NLG, one needs a way of representing *all* the information that is needed for generating a text, and this usually has complex internal structure.

Linguistic ontologies are ontologies developed to describe linguistic concepts. Although ontologies are used in a number of NLP projects (e.g. (Estival et al., 2004)), the ontologies used are usually ontologies of the application domain rather

than the linguistic structures of natural languages. The development of ontologies to describe aspects of natural languages is comparatively rare. The WordNet ontologies are a widely used resource describing the repertoire of word senses of natural languages, but these concentrate on individual words rather than larger linguistic structures. More relevant to NLG is work on various versions of the Generalised Upper Model (Bateman et al., 1995), which outlines aspects of meaning relevant to making NLG decisions. This has been used to help translate domain knowledge in a number of NLG systems (Aguado et al., 1998).

In summary, existing approaches to using ontologies or XML for natural language related purposes are not adequate to describe the datastructures needed for NLG. Semantic web technology applied to specifications with the complexity of those generated by RAGS might, however, be able to fill this gap.

## 9 The Semantic Web for NLG tasks

In the above, we have made a case for the use of semantic web technology to aid inter-operability and sharing of resources for NLG. This was justified largely by the fact that the most significant NLG “standardisation” effort so far, RAGS, can be straightforwardly recast in semantic web terms, bringing distinct advantages. Even if RAGS itself is not taken forward in its current form, this suggests that further developments of the idea could bear fruit in semantic web terms.

The semantic web is certainly not a panacea for all the problems of NLG, and indeed there are aspects of the technology that are still at an early stage of development. For instance, the problems of matching/reconciling alternative ontologies are many and complex. Some researchers even dispute the viability of the general approach. On the other hand, the semantic web community is concerned with a number of problems that are also very relevant to NLG. Fundamentally, the semantic web is about sharing and exploiting distributed computational resources in an open community where many different goals, viewpoints and theories are represented. This is something that NLG also seeks to do in a number of ways. The semantic web movement has considerable momentum. There are more of them than us. Let’s see what we can get from it.

## Acknowledgments

This work was supported by EPSRC grant EP/E011764/1.

## References

- G. Aguado, A. Ba n3n, John A. Bateman, S. Bernardos, M. Fern3ndez, A. G3mez-P3rez, E. Nieto, A. Olalla, R. Plaza, and A. S3nchez. 1998. Ontogeneration: Reusing domain and linguistic ontologies for Spanish text generation. In *Proceedings of the ECAI'98 Workshop on Applications of Ontologies and Problem Solving Methods*, pages 1–10, Brighton, UK.
- Grigoris Antoniou and Frank van Harmelen. 2004. *A Semantic Web Primer*. MIT Press.
- John A. Bateman, Renate Henschel, and Fabio Rinaldi. 1995. Generalized Upper Model 2.0: documentation. Technical report, GMD/Institut f3ur Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.
- John Bateman, Renate Henschel, and Judy Delin. 2002. A brief introduction to the GeM annotation scheme for complex document layout. In *Proceedings of NLP-XML 2002*, Taipei.
- D. Brickley and R. V. Guha. 2003. Rdf vocabulary description language 1.0: Rdf schema. Technical Report <http://www.w3.org/TR/rdf-schema>, World Wide Web Consortium.
- Lynne Cahill, Roger Evans, Chris Mellish, Daniel Paiva, Mike Reape, and Donia Scott. 2001. The RAGS Reference Manual . Available at <http://www.itri.brighton.ac.uk/projects/rags>.
- Dominique Estival, Chris Nowak, and Andrew Zschorn. 2004. Towards ontology-based natural language processing. In *Proceedings of NLP-XML 2004*, Barcelona.
- Scott Farrar and Terry Langendoen. 2003. A linguistic ontology for the semantic web. *Glott International*, 7(3):1–4.
- Ian Horrocks. 2005. Description logics in ontology applications. In B. Beckert, editor, *Proc. of the 9th Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2005)*, pages 2–13. Springer Verlag LNCS 3702.
- Baden Hughes and Steven Bird. 2003. Grid-enabling natural language engineering by stealth. In *Proceedings of the HLT-NAACL 2003 Workshop on The Software Engineering and Architecture of Language Technology Systems*.
- G. Klyne and J. Carroll. 2003. Resource description framework (rdf): Concepts and abstract syntax. Technical Report <http://www.w3.org/TR/rdf-concepts>, World Wide Web Consortium.
- D. L. McGuinness and F. van Harmelen. 2004. Owl web ontology language overview. <http://www.w3.org/TR/owl-features/>.
- Chris Mellish, Donia Scott, Lynne Cahill, Daniel Paiva, Roger Evans, and Mike Reape. 2006. A reference architecture for generation systems. *Natural language engineering*, 1:1–34.
- G. Miller. 1995. Wordnet: A lexical database for english. *CACM*, 38(11):39–41.
- Richard Power, Donia Scott, and Nadjet Bouayad-Agha. 2003. Document structure. *Computational Linguistics*, 29:211–260.
- Stuart M. Shieber. 1986. *An introduction to unification-based approaches to grammar*. CSLI.



## Natural Reference to Objects in a Visual Domain

**Margaret Mitchell**  
 Computing Science Dept.  
 University of Aberdeen  
 Scotland, U.K.

**Kees van Deemter**  
 Computing Science Dept.  
 University of Aberdeen  
 Scotland, U.K.

**Ehud Reiter**  
 Computing Science Dept.  
 University of Aberdeen  
 Scotland, U.K.

{m.mitchell, k.vdeemter, e.reiter}@abdn.ac.uk

### Abstract

This paper discusses the basic structures necessary for the generation of reference to objects in a visual scene. We construct a study designed to elicit naturalistic referring expressions to relatively complex objects, and find aspects of reference that have not been accounted for in work on Referring Expression Generation (REG). This includes reference to object parts, size comparisons without crisp measurements, and the use of analogies. By drawing on research in cognitive science, neurophysiology, and psycholinguistics, we begin developing the input structure and background knowledge necessary for an algorithm capable of generating the kinds of reference we observe.

### 1 Introduction

One of the dominating tasks in Natural Language Generation (NLG) is the generation of expressions to pick out a referent. In recent years there has been increased interest in generating referential expressions that are *natural*, e.g., like those produced by people. Although research on the generation of referring expressions has examined different aspects of how people generate reference, there has been surprisingly little research on how people refer to objects in a real-world setting. This paper addresses this issue, and we begin formulating the requirements for an REG algorithm that refers to visible three-dimensional objects in the real world.

Reference to objects in a visual domain provides a straightforward extension of the sorts of reference REG research already tends to consider. Toy examples outline reference to objects, people, and animals that are perceptually available before the speaker begins generating an utterance (Dale and Reiter, 1995; Krahmer et al., 2003; van

Deemter et al., 2006; Areces et al., 2008). Example referents may be referred to by their color, size, type (“dog” or “cup”), whether or not they have a beard, etc.

Typically, the reference process proceeds by comparing the properties of the referent with the properties of all the other items in the set. The final expression roughly conforms to the Gricean maxims (Grice, 1975).

However, when the goal is to generate natural reference, this framework is too simple. The form reference takes is profoundly affected by modality, task, and audience (Chapanis et al., 1977; Cohen, 1984; Clark and Wilkes-Gibbs, 1986), and even when these aspects are controlled, different people will refer differently to the same object (Mitchell, 2008). In light of this, we isolate one kind of natural reference and begin building the algorithmic framework necessary to generate the observed language.

Psycholinguistic research has examined reference in a variety of settings, which may inform research on natural REG, but it is not always clear how to extend this work to a computational model. This is true in part because these studies favor an analysis of reference in the context of collaboration; reference is embedded within language, and language is often a joint activity. However, most research on referring expression generation supposes a solitary generating agent.<sup>1</sup> This tacitly assumes that reference will be taking place in a monologue setting, rather than a dialogue or group setting. Indeed, the goal of most REG algorithms is to produce uniquely distinguishing, one-shot referring expressions.

Studies on natural reference usually use a two person (speaker-listener) communication task (e.g., Flavell et al., 1968; Krauss and Glucksberg, 1969; Ford and Olson, 1975). This research has

<sup>1</sup>A notable exception is Heeman and Hirst (1995).

shown that reference is more accurate and efficient when it incorporates things like gesture and gaze (Clark and Krych, 2004). There is a trade-off in effort between initiating a noun phrase and refashioning it so that both speakers understand the referent (Clark and Wilkes-Gibbs, 1986), and speakers communicate to form lexical pacts on how to refer to an object (Sacks and Schegloff, 1979; Brennan and Clark, 1996). Mutual understanding of referents is achieved in part by referring within a subset of potential referents (Clark et al., 1983; Beun and Cremers, 1998). A few studies have compared monologue to dialogue reference, and have shown that monologue references tend to be harder for a later listener to disambiguate (Clark and Krych, 2004) and that subsequent references tend to be longer than those in dialogues (Krauss and Weinheimer, 1967).

Aiming to generate natural reference in a monologue setting raises questions about what an algorithm should use to produce utterances like those produced by people. In a monologue setting, the speaker (or algorithm) gets no feedback from the listener; the speaker's reference is not tied to interactions with other participants. The speaker is therefore in a difficult position, attempting to clearly convey a referent without being able to check if the reference is understood along the way.

Recent studies that have focused on monologue reference do so rather explicitly, which may affect participant responses. These studies utilize 2D graphical depictions of simple 3D objects (van Deemter et al., 2006; Viethen and Dale, 2008), where a small set of properties can be used to distinguish one item from another. The expressions are elicited in isolation, typed and then submitted, which may hide some of the underlying referential processes. None of these studies utilize actual objects. It is therefore difficult to use these data to draw conclusions about how reference works in naturalistic settings. It is unclear if these experimental settings are natural enough, i.e., if they get at reference as it may occur every day.

The study in this paper attempts to bring out information about reference in a number of ways. First, we conduct the study in-person, using real-world objects. This design invites referential phenomena that may not have been previously observed in simpler domains. Second, the referring expressions are produced orally. This allows us access to reference as it is generated, without

the participants revising and so potentially obscuring information about their reference. Third, we use a relatively complicated task, where participants must explain how to use pieces to put together a picture of a face. The fact that we are looking at reference is not made explicit, which lessens any experimental effects caused by subjects guessing the purpose of the study. This approach also situates reference within a larger task, which may draw out aspects of reference not usually seen in experiments that elicit reference in isolation. Fourth, the objects used display a variety of different features: texture, material, color, size along several dimensions, etc. This brings the data set closer to objects that people interact with every day. A monologue setting offers a picture of the phenomena at play during a single individual's referring expression generation.

The referring expressions gathered in this study exhibit several aspects of reference that have not yet been addressed in REG. This includes (1) part-whole modularity; (2) size comparisons across three dimensions; and (3) analogies. Work in cognitive sciences suggests that these phenomena are interrelated, and may be possible to represent in a computational framework. This research also offers connections to further aspects of natural reference that were not directly observed in the study, but will need to be accounted for in future work on naturalistic referring expression generation. Using these ideas, we begin formulating the structures that an REG algorithm would need in order to produce reference to real-world objects in a visual setting.

Approaching REG in this way allows us to tie research in the generation of referring expressions to computational models of visual perception and cognitively-motivated computer vision. Moving in this direction offers the prospect of eventually developing an application for the generation of natural reference to objects automatically recognized by a computer vision system.

In the next section, we describe our study. In Section 3, we analyze the results and discuss what they tell us about natural reference. In Section 4, we draw on our results and cognitive models of object recognition to begin building the framework for a referring expression algorithm that generates naturalistic reference to objects in a visual scene. In Section 5, we offer concluding remarks and outline areas for further study.



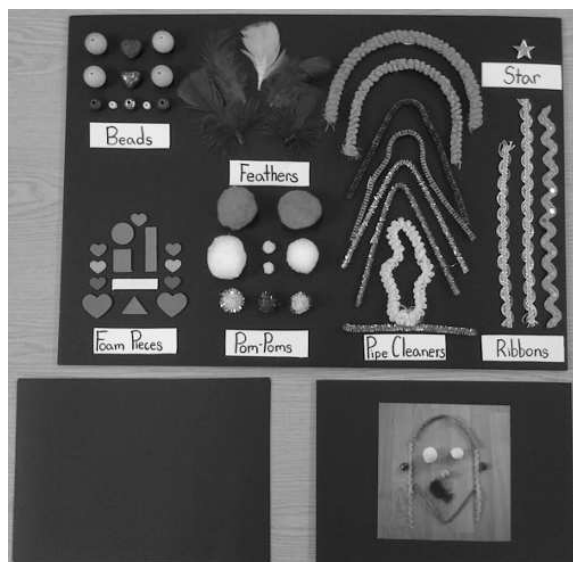


Figure 1: Object Board.

## 2 Method

### 2.1 Subjects

The subjects were 20 residents of Aberdeen, Scotland, and included undergraduates, graduates, and professionals. All were native speakers of English, had normal or corrected vision, and had no other known visual issues (such as color-blindness). Subjects were paid for their participation. Two recordings were left out of the analysis: one participant's session was not fully recorded due to a software error, and one participant did not pick out many objects in each face and so was not included. The final set of participants included 18 people, 10 female and 8 male.

### 2.2 Materials

A board was prepared with 51 craft objects. The objects were chosen from various craft sets, and included pom-poms, pipe-cleaners, beads, and feathers (see Table 1). The motley group of objects had different colors, textures, shapes, patterns, and were made of different materials. Similar objects were grouped together on the board, with a label placed underneath. This was done to control the head noun used in each reference. The objects were used to make up 5 different craft "face" pictures. Subjects sat at a desk facing the board and the stack of pictures. A picture of the board is shown in Figure 1.

Subjects were recorded on a head-mounted microphone, which fed directly into a laptop placed on the left of the desk. The open-source audio-

recording program Audacity (Mazzoni, 2010) was used to record the audio signal and export it to wave format.

### 2.3 Procedure

Subjects were told to give instructions on how to construct each face using the craft supplies on the board. They were instructed to be clear enough for a listener to be able to reconstruct each face without the pictures, with only the board items in front of them. A pilot study revealed that such open-ended instructions left some subjects spending an inordinate amount of time on the exact placement of each piece, and so in the current study subjects were told that each face should take "a couple" minutes, and that the instructions should be as clear as possible for a listener to use the same objects in reconstructing the pictures without being "overly concerned" with the details of exactly how each piece is angled in relation to the other.

Subjects were first given a practice face to describe. This face was the same face for all subjects. They were then allowed to voice any concerns or ask questions, but the experimenter only repeated portions of the original instructions; no new information was given. The subject could then proceed to the next four faces, which were in a random order for each subject. A transcript of a single face from a session is provided in Figure 2.

### 2.4 Analysis

The recordings of each monologue were transcribed, including disfluencies, and each face section ("eyes", "chin", etc.) was marked. First reference to items on the board were annotated with their corresponding item numbers, yielding 722 references.<sup>2</sup> Initial references to single objects were extracted, creating a final data set with 505 references to single objects.

## 3 Results

Each reference was annotated in terms of the properties used to pick out the referent. For example, "the red feather" was annotated as containing the <ATTRIBUTE:value> pairs <COLOR:red, TYPE:feather>. Discerning properties from the modifiers used in reference is generally straightforward, and all of the references produced may be partially deconstructed using such properties.

<sup>2</sup>This corpus is available at [http://www.csd.abdn.ac.uk/~mitchema/craft\\_corpus](http://www.csd.abdn.ac.uk/~mitchema/craft_corpus).

<b>14 foam shapes</b>	2 large red hearts	2 small red hearts	2 small neon green hearts
2 small blue hearts	1 small green heart	1 green triangle	1 red circle
1 red square	1 red rectangle	1 white rectangle	
<b>11 beads</b>	4 large round wooden beads	2 small white plastic beads	2 brown patterned beads
1 gold patterned bead	1 shiny gold patterned heart	1 red patterned heart	
<b>9 pom poms</b>	2 big green pom-poms	2 small neon green pom-poms	2 small silver pom-poms
1 small metallic green pom-pom		1 large white pom-pom	1 medium white pom-pom
<b>8 pipe cleaners</b>	1 gold pipe-cleaner	1 gold pipe-cleaner in half	1 silver pipe-cleaner
1 circular neon yellow soft pipe-cleaner		1 neon orange puffy pipe-cleaner	1 grey puffy pipe-cleaner
1 purple/yellow striped pipe-cleaner		1 brown/grey striped pipe-cleaner	
<b>5 feathers</b>	2 purple feathers	2 red feathers	1 yellow feather
<b>3 ribbons</b>	1 gold sequined wavy ribbon	1 silver wavy ribbon	1 small silver wavy ribbon
<b>1 star</b>	1 gold star		

Table 1: Board items.

<CHIN> Okay so this face again um this face has um uh for the chin, it uses (10 *a gold pipe-cleaner in a V shape*) where the bottom of the V is the chin. </CHIN>  
 <MOUTH> The mouth is made up of (9 *a purple feather*). And the mouth is slightly squint, um as if the the person is smiling or even smirking. So this this smile is almost off to one side. </MOUTH>  
 <NOSE> The nose is uh (5 *a wooden bead, a medium-sized wooden bead with a hole in the center*). </NOSE>  
 <EYES> And the eyes are made of (2,3 *white pom-poms*), em just uh em evenly spaced in the center of the face. </EYES>  
 <FOREHEAD> Em it's see the person's em top of the person's head is made out of (1 *another, thicker pipe-cleaner that's uh a grey color, it's kind of uh a knotted blue-type pipe-cleaner*). So that that acts as the top of the person's head. </FOREHEAD>  
 <HAIR> And down the side of the person's face, there are (7,8 *two ribbons*) on each side. (7,8 *And those are silver ribbons*). Um and they just hang down the side of the face and they join up the the grey pipe-cleaner and the top um of the person's head to the to the chin and then hang down either side of the chin. </HAIR>  
 <EARS> And the person's ears are made up of (4,6 *two beads, which are um love-heart-shaped beads*), where the points of the love-hearts are facing outwards. And those are just placed um around same em same em horizontal line as the nose of the person's face is. </EARS>

Figure 2: Excerpt Transcript.

Using sets of properties to distinguish referents is nothing new in REG. Algorithms for the generation of referring expressions commonly use this as a starting point, proposing that properties are organized in some linear order (Dale and Reiter, 1995) or weighted order (Krahmer et al., 2003) as input. However, we find evidence that more is at play. A breakdown of our findings is listed in Table 2.

### 3.1 Spatial Reference

In addition to properties that pick out referents, throughout the data we see reference to objects as they exist in space. Size is compared across different dimensions of different objects, and reference is made to different parts of the objects,

picking out pieces within the whole. These two phenomena – relative size comparisons and part-whole modularity – point to an underlying spatial object representation that may be utilized during reference.

#### 3.1.1 Relative Size Comparisons

A total of 122 (24.2%) references mention size with a vague modifier (e.g., “big”, “wide”). This includes comparative (e.g., “larger”) and superlative (e.g., “largest”) size modifiers, which occur 40 (7.9%) times in the data set. Examples are given below.

- (1) “*the bigger pom-pom*”
- (2) “*the green largest pom-pom*”
- (3) “*the smallest long ribbon*”
- (4) “*the large orange pipe-cleaner*”

Of the references that mention size, 35 (6.9%) use a vague modifier that applies to one or two dimensions. This includes modifiers for height (“the short silver ribbon”), width (“quite a fat rectangle”), and depth (“the thick grey pipe-cleaner”). 87 (17.2%) use a modifier that applies to the overall size of the object (e.g., “big” or “small”). Table 3 lists these values. Crisp measurements (such as “1 centimeter”) occur only twice (0.4%), with both produced by the same participant.

Comparative/Superlative:	40 (7.9%)
Base:	82 (16.2%)
Height/Width/Depth:	35 (6.9%)
Overall size:	87 (17.2%)

Table 3: Size Modifier Breakdown.

Part-whole modularity	Relative size	Analogies
“a green pom-pom... with the tinsel on the outside” “your gold twisty ribbon... with sequins on it” “a wooden bead... with a hole in the center” “one of the green pom-poms... with the sort of strands coming out from it.” “the silver ribbon... with the chainmail detail down through the middle of it.”	“a red foam-piece... which is more square in shape rather than the longer rectangle” “the grey pipe-cleaner... which is the thicker one... “the slightly larger one” “the smaller silver ribbon” “the short silver ribbon” “quite a fat rectangle” “thick grey pipe-cleaner”	“a natural-looking piece of pipe-cleaner, it looks a bit like a rope” “a pipe-cleaner that looks a bit like... a fluffy caterpillar” “the silver ribbon that’s almost like a big S shape.” “a... pipe-cleaner that looks like tinsel.”
11 References	122 References	16 References

Table 2: Examples of Observed Reference.

Participants produce such modifiers without sizes or measurements explicitly given; with an input of a visual object presentation, the output includes size modifiers. Such data suggests that natural reference in a visual domain utilizes processes comparing the length, width, and height of a target object with other objects in the set. Indeed, 5 references (1.0%) in our data set include explicit comparison with the size of other objects.

- (5) *“a red foam-piece... which is more square in shape rather than the longer rectangle”*
- (6) *“the grey pipe-cleaner... which is the thicker one... of the selection”*
- (7) *“the shorter of the two silver ribbons”*
- (8) *“the longer one of the ribbons”*
- (9) *“the longer of the two silver ribbons”*

In Example (5), height and width across two different objects are compared, distinguishing a square from a rectangle. In (6) “thicker” marks the referent as having a larger circumference than other items of the same type. (7) (8) and (9) compare the height of the target referent to the height of similar items.

The use of size modifiers in a domain without specified measurements suggests that when people refer to an object in a visual domain, they are sensitive to its size and structure within a dimensional, real-world space. Without access to crisp measurements, people compare relative size across different objects, and this is reflected in the expressions they generate. These comparisons are not only limited to overall size, but include size in each dimension. This suggests that objects’ structures within a real-world space are relevant to REG in a visual domain.

### 3.1.2 Part-Whole Modularity

The role that a spatial object understanding has within reference is further detailed by utterances that pick out the target object by mentioning an object part. 11 utterances (2.2%) in our data include mention of an object part within reference to the whole object. This is spread across participants, such that half of the participants make reference to an object part at least once.

- (10) *“a green pom-pom, which is with the tinsel on the outside”*
- (11) *“your gold twisty ribbon...with sequins on it”*
- (12) *“a wooden bead...with a hole in the center”*

In (10), pieces of tinsel are isolated from the whole object and specified as being on the outside. In (11), smaller pieces that lay on top of the ribbon are picked out. And in (12), a hole within the bead is isolated.

The use of part-whole modularity suggests an understanding that parts of the object take up their own space within the object. An object is not only viewed as a whole during reference, but parts in, on, and around it may be considered as well. For an REG algorithm to generate these kinds of references, it must be provided with a representation that details the structure of each object.

## 3.2 ANALOGIES

The data from this study also provide information on what can be expected from a knowledge base in an algorithm that aims to generate naturalistic reference. Reference is made 16 times (3.2%) to objects not on the board, where the intended referent is compared against something it is *like*. Some examples are given below.

- (13) *“a gold...pipe-cleaner... completely straight, like a ruler”*
- (14) *“a natural-looking piece of pipe-cleaner, it looks a bit like a rope”*
- (15) *“a pipe-cleaner that looks a bit like... a fluffy caterpillar...”*

In (13), a participant makes reference to a SHAPE property of an object not on the board. In (14) and (15), participants refer to objects that may share a variety of properties with the referent, but are also not on the board.

Reference to these other items do not pick out single objects, but types of objects (e.g., an object *type*, not *token*). They correspond to some prototypical idea of an object with properties similar to those of the referent. Work by Rosch (1975) has examined this tendency, introducing the idea of *prototype theory*, which proposes that there may be some central, ‘prototypical’ notions of items. A knowledge base with stored prototypes could be utilized by an REG algorithm to compare the target referent to item prototypes. Such representations would help guide the generation of reference to items not in the scene, but similar to the target referent.

#### 4 Discussion

We have discussed several different aspects of reference in a study where referring expressions are elicited for objects in a spatial, visual scene. Reference in this domain draws on object forms as they exist in a three-dimensional space and utilizes background knowledge to describe referents by analogy to items outside of the scene. This is undoubtedly not an exhaustive account of the phenomena at play in such a domain, but offers some initial conclusions that may be drawn from exploratory work of this kind.

Before continuing with the discussion, it is worthwhile to consider whether some of our data might be seen as going beyond reference. Perhaps the participants are doing something else, which could be called describing. How to draw the line between a distinguishing reference and a description, and whether such a line can be drawn at all, is an interesting question. If the two are clearly distinct, then both are interesting to NLG research. If the two are one in the same, then this sheds some light on how REG algorithms should treat

reference. We leave a more detailed discussion of this for future work, but note recent psycholinguistic work suggesting that referring establishes (1) an individual as the referent; (2) a conceptualization or perspective on that individual (Clark and Bangerter, 2004). Schematically, referring = indicating + describing.

We now turn to a discussion of how the observed phenomena may be best represented in an REG algorithm. We propose that an algorithm capable of generating natural reference to objects in a visual scene should utilize (1) a spatial object representation; (2) a non-spatial feature-based representation; and (3) a knowledge base of object prototypes.

##### 4.1 Spatial and Visual Properties

It is perhaps unsurprising to find reference that exhibits spatial knowledge in a study where objects are presented in three-dimensional space. Human behavior is anchored in space, and spatial information is essential for our ability to navigate the world we live in. However, referring expression generation algorithms geared towards spatial representations have oversimplified this tendency, keeping objects within the realm of two-dimensions and only looking at the spatial relations between objects.

For example, Funakoshi et al. (2004) and Gatt (2006) focus on how objects should be clustered together to form groups. This utilizes some of the spatial information between objects, but does not address the spatial, three-dimensional nature of objects themselves. Rather, objects exist as entities that may be grouped with other entities in a set or singled out as individual objects; they do not have their own spatial characteristics. Similarly, one of the strengths of the Graph-Based Algorithm (Krahmer et al., 2003) is its ability to generate expressions that involve relations between objects, and these include spatial ones (“next to”, “on top of”, etc.). In all these approaches, however, objects are essentially one-dimensional, represented as individual nodes.

Work that does look at the spatial information of different objects is provided by Kelleher et al. (2005). In this approach, the overall volume of each object is calculated to assign salience rankings, which then allow the Incremental Algorithm (Dale and Reiter, 1995) to produce otherwise “underspecified” reference. The spatial properties of

the objects are kept relatively simple. They are not used in constructing the referring expression, but one aspect of the object's three-dimensional shape (volume) affects the referring expression's final form. To the authors' knowledge, the current work is the first to suggest that objects themselves should have their spatial properties represented during reference.

Research in cognitive modelling supports the idea that we attend to the spatial properties of objects when we view them (Blaser et al., 2000), and that we have purely spatial attentional mechanisms operating alongside non-spatial, feature-based attentional mechanisms (Treue and Trujillo, 1999). These feature-based attentional mechanisms pick out properties commonly utilized in REG, such as texture, orientation, and color. They also pick out edges and corners, contrast, and brightness. Spatial attentional mechanisms provide information about where the non-spatial features are located in relation to one another, size, and the spatial interrelations between component parts.

Applying these findings to our study, an REG algorithm that generates natural reference should utilize a visual, feature-based representation of objects alongside a structural, spatial representation of objects. A feature-based representation is already common to REG, and could be represented as a series of <ATTRIBUTE:value> pairs. A spatial representation is necessary to define how the object is situated within a dimensional space, providing information about the relative distances between object components, edges, and corners.

With such information provided by a spatial representation, the generation of part-whole expressions, such as “the pom-pom with the tinsel on the outside”, is possible. This also allows for the generation of size modifiers (“big”, “small”) without the need for crisp measurements, for example, by comparing the difference in overall height of the target object with other objects in the scene, or against a stored prototype (discussed below). Relative size comparisons across different dimensions would also be possible, used to generate size modifiers such as “wide” and “thick” that refer to one dimensional axis.

## 4.2 Analogies

A feature-based and a spatial representation may also play a role in analogies. When we use analogies, as in “the pipe-cleaner that looks like a cater-

pillar”, we use world knowledge about items that are not themselves visible. Such an expression draws on similarity that does not link the referent with a particular object, but with a general type of object: the pipe-cleaner is caterpillar-like.

To generate these kinds of expressions, an REG algorithm would first need a knowledge base with prototypes listing prototypical values of attributes. For example, a banana prototype might have a prototypical COLOR of yellow. With prototypes in the knowledge base, the REG algorithm would need to calculate similarity of a target referent to other known items. This would allow a piece of yellow cloth, for example, to be described as being the color of a banana.

Implementing such similarity measures in an REG algorithm will be challenging. One difficulty is that prototype values may be different depending on what is known about an item; a prototypical unripe banana may be green, or a prototypical rotten banana brown. Another difficulty will be in determining when a referent is similar *enough* to a prototype to warrant an analogy. Additional research is needed to explore how these properties can be reasoned about.

## 4.3 Further Implications

A knowledge base containing prototypes opens up the possibility of generating many other kinds of natural references. In particular, such knowledge would allow the algorithm to compute which properties a given kind of referent may be expected to have, and which properties may be unexpected. Unexpected properties may therefore stand out as particularly salient.

For example, a dog missing a leg may be described as a “three-legged dog” because the prototypical dog has four legs. We believe that this perspective, which hinges on the unexpectedness of a property, suggests a new approach to attribute selection. Unlike the Incremental Algorithm, the Preference Order that determines the order in which attributes are examined would not be fixed, but would depend on the nature of the referent and what is known about it.

Approaching REG in this way follows work in cognitive science and neurophysiology that suggests that expectations about objects' visual and spatial characteristics are derived from stored representations of object 'prototypes' in the inferior temporal lobe of the brain (Logothetis and

- A spatial representation (depicting size, inter-relations between component parts)
- A non-spatial, propositional representation (describing color, texture, orientation, etc.)
- A knowledge base with stored prototypical object propositional and spatial representations

Table 4: Requirements for an REG algorithm that generates natural reference to visual objects.

Sheinberg, 1996; Riesenhuber and Poggio, 2000; Palmeri and Gauthier, 2004). Most formal theories of object perception posit some sort of *category activation system* (Kosslyn, 1994), a system that matches input properties of objects to those of stored prototypes, which then helps guide expectations about objects in a top-down fashion.<sup>3</sup> This appears to be a neurological correlate of the knowledge base we propose to underlie analogies.

Such a system contains information about prototypical objects' component parts and where they are placed relative to one another, as well as relevant values for material, color, etc. This suggests that the spatial and non-spatial feature-based representations proposed for visible objects could be used to represent prototype objects as well. Indeed, how we view and refer to objects appears to be influenced by the interaction of these structures: Expectations about an object's spatial properties guide our attention towards expected object parts and non-spatial, feature-based properties throughout the scene (Kosslyn, 1994; Itti and Koch, 2001). This affects the kinds of things we are most likely to generate language about (Itti and Arbib, 2005).

We can now outline some general requirements for an algorithm capable of generating naturalistic reference to objects in a visual scene: Input to such an algorithm should include a feature-based representation, which we will call a *propositional representation*, with values for color, texture, etc., and a *spatial representation*, with symbolic information about objects' size and the spatial relationships between components. A system that generates naturalistic reference must also use a knowledge base storing information about object prototypes, which may be represented in terms of their own propositional/spatial representations.

<sup>3</sup>Note that this is not the only proposed matching structure in the brain – an *exemplar activation system* matches input to stored exemplars.

## 5 Conclusions and Future Work

We have explored the interaction between viewing objects in a three-dimensional, spatial domain and referring expression generation. This has led us to propose structures that may be used to connect vision in a spatial modality to naturalistic reference. The proposed structures include a spatial representation, a propositional representation, and a knowledge base with representations for object prototypes. Using structures that define the propositional and spatial content of objects fits well with work in psycholinguistics, cognitive science and neurophysiology, and may provide the basis to generate a variety of natural-sounding references from a system that recognizes objects.

It is important to note that any naturalistic experimental design limits the kinds of conclusions that can be drawn about reference. A study that elicits reference to objects in a visual scene provides insight into reference to objects in a visual scene; these conclusions cannot easily be extended to reference to other kinds of phenomena, such as reference to people in a novel. We therefore make no claims about reference as a whole in this paper; generalizations from this research can provide hypotheses for further testing in different modalities and with different sorts of referents.

Our data leave open many areas for further study, and we hope to address these in future work. Experiments designed specifically to elicit relative size modifiers, reference to object components, and reference to objects that are *like* other things would help further detail the form our proposed structures take.

What is clear from our data is that both a spatial understanding and a non-spatial feature-based understanding appear to play a role in reference to objects in a visual scene, and further, reference in such a setting is bolstered by a knowledge base with stored prototypical object representations. Utilizing structures representative of these phenomena, we may be able to extend object recognition research into object reference research, generating natural-sounding reference in everyday settings.

## Acknowledgements

Thanks to Advait Siddharthan for thought-provoking discussions and to the anonymous reviewers for useful suggestions.

## References

- Carlos Areces, Alexander Koller, and Kristina Striegnitz. 2008. Referring expressions as formulas of description logic. *Proceedings of the Fifth International Natural Language Generation Conference*, pages 42–29.
- Robbert-Jan Beun and Anita H. M. Cremers. 1998. Object reference in a shared domain of conversation. *Pragmatics and Cognition*, 6:121–52.
- Erik Blaser, Zenon W. Pylyshyn, and Alex O. Holcombe. 2000. Tracking an object through feature space. *Nature*, 408:196–199.
- Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:1482–93.
- Alphonse Chapanis, Robert N. Parrish, Robert B. Ochsman, and Gerald D. Weeks. 1977. Studies in interactive communication: II. the effects of four communication modes on the linguistic performance of teams during cooperative problem solving. *Human Factors*, 19:101–125.
- Herbert H. Clark and Adrian Bangerter. 2004. Changing ideas about reference. In Ira A. Noveck and Dan Sperber, editors, *Experimental pragmatics*, pages 25–49. Palgrave Macmillan, Basingstoke, England.
- Herbert H. Clark and Meredyth A. Krych. 2004. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50:62–81.
- Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.
- Herbert H. Clark, Robert Schreuder, and Samuel Buttrick. 1983. Common ground and the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, 22:1–39.
- Philip R. Cohen. 1984. The pragmatics of referring and the modality of communication. *Computational Linguistics*, 10(2):97–146.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.
- J. H. Flavell, P. T. Botkin, D. L. Fry Jr., J. W. Wright, and P. E. Jarvice. 1968. *The Development of Role-Taking and Communication Skills in Children*. John Wiley, New York.
- William Ford and David Olson. 1975. The elaboration of the noun phrase in children’s description of objects. *The Journal of Experimental Child Psychology*, 19:371–382.
- Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. 2004. Generating referring expressions using perceptual groups. In *Proceedings of the 3rd International Conference on Natural Language Generation*, pages 51–60.
- Albert Gatt. 2006. Structuring knowledge for reference generation: A clustering algorithm. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 321–328.
- Paul H. Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:41–58.
- Peter A. Heeman and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21.
- Laurent Itti and Michael A. Arbib. 2005. Attention and the minimal subscene. In Michael A. Arbib, editor, *Action to Language via the Mirror Neuron System*. Cambridge University Press.
- Laurent Itti and Christof Koch. 2001. Computational modelling of visual attention. *Nature Reviews Neuroscience*.
- J. Kelleher, F. Costello, and J. van Genabith. 2005. Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *Artificial Intelligence*, 167:62–102.
- Stephen M. Kosslyn. 1994. *Image and Brain: The Resolution of the Imagery Debate*. MIT Press, Cambridge, MA.
- Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- Robert M. Krauss and Sam Glucksberg. 1969. The development of communication: Competence as a function of age. *Child Development*, 40:255–266.
- Robert M. Krauss and Sidney Weinheimer. 1967. Effect of referent similarity and communication mode on verbal encoding. *Journal of Verbal Learning and Verbal Behavior*, 6:359–363.
- Nikos K. Logothetis and David L. Sheinberg. 1996. Visual object recognition. *Annual Review Neuroscience*, 19:577–621.
- Dominic Mazzoni. 2010. Audacity.
- Margaret Mitchell. 2008. Towards the generation of natural reference. Master’s thesis, University of Washington.
- Thomas J. Palmeri and Isabel Gauthier. 2004. Visual object understanding. *Nature Reviews Neuroscience*, 5:291–303.
- Maximilian Riesenhuber and Tomaso Poggio. 2000. Models of object recognition. *Nature Neuroscience Supplement*, 3:1199–1204.

Eleanor Rosch. 1975. Cognitive representation of semantic categories. *Journal of Experimental Psychology*, 104:192–233.

Harvey Sacks and Emanuel A. Schegloff. 1979. Two preferences in the organization of reference to persons in conversation and their interaction. In George Psathas, editor, *Everyday Language: Studies in Ethnomethodology*, pages 15–21. Irvington Publishers, New York.

Stegan Treue and Julio C. Martinez Trujillo. 1999. Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399:575–579.

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Conference on Natural Language Generation*, Sydney, Australia. ACL.

Jette Viethen and Robert Dale. 2008. The use of spatial descriptions in referring expressions. In *Proceedings of the 5th International Conference on Natural Language Generation, INLG-08*, Salt Fork, Ohio. ACL.



# Generating and Validating Abstracts of Meeting Conversations: a User Study

Gabriel Murray

gabrielm@cs.ubc.ca

Giuseppe Carenini

carenini@cs.ubc.ca

Raymond Ng

rng@cs.ubc.ca

Department of Computer Science, University of British Columbia  
Vancouver, Canada

## Abstract

In this paper we present a complete system for automatically generating natural language abstracts of meeting conversations. This system is comprised of components relating to *interpretation* of the meeting documents according to a meeting ontology, *transformation* or *content selection* from that source representation to a summary representation, and *generation* of new summary text. In a formative user study, we compare this approach to gold-standard human abstracts and extracts to gauge the usefulness of the different summary types for browsing meeting conversations. We find that our automatically generated summaries are ranked significantly higher than human-selected extracts on coherence and usability criteria. More generally, users demonstrate a strong preference for abstract-style summaries over extracts.

## 1 Introduction

The most common solution to the task of summarizing spoken and written data is sentence (or utterance) extraction, where binary sentence classification yields a cut-and-paste summary comprising informative sentences from the document concatenated in a new, condensed document. Such extractive approaches have dominated the field of automatic summarization for decades, in large part because extractive systems do not require a natural language generation (NLG) component since the summary sentences are simply lifted from the source document.

Extrinsic evaluations have shown that, while extractive summaries may be less coherent than human abstracts, users still find them to be valuable tools for browsing documents (He et al., 1999; McKeown et al., 2005; Murray et al., 2009). However, these previous evaluations also illustrate that

concise abstracts are generally preferred by users and lead to higher objective task scores. A weakness of typical extractive summaries is that the end user does not know *why* the extracted sentences are important; exploring the original sentence context may be the only way to resolve this uncertainty. And if the input source document consists of noisy, unstructured text such as ungrammatical, disfluent multi-party speech, then the resultant extract is likely to be noisy and unstructured as well.

Herein we describe a complete and fully automatic system for generating abstract summaries of meeting conversations. Our abstractor maps input sentences to a meeting ontology, generates *messages* that abstract over multiple sentences, selects the most informative messages, and ultimately generates new text to describe these relevant messages at a high level. We conduct a user study where participants must browse a meeting conversation within a very constrained timeframe, having a summary at their disposal. We compare our automatic abstracts with human abstracts and extracts and find that our abstract summaries significantly outperform extracts in terms of coherence and usability according to human ratings. In general, users rate abstract-style summaries much more highly than extracts for these conversations.

## 2 Related Research

Automatic summarization has been described as consisting of *interpretation*, *transformation* and *generation* (Jones, 1999). Popular approaches to text extraction essentially collapse interpretation and transformation into one step, with generation either being ignored or consisting of post-processing techniques such as sentence compression (Knight and Marcu, 2000; Clarke and Lapata, 2006) or sentence merging (Barzilay and McKeown, 2005). In contrast, in this work we clearly separate interpretation from transformation and incorporate an NLG component to generate new text to describe meeting conversations.

While extraction remains the most common ap-

proach to text summarization, one application in which abstractive summarization is widely used is data-to-text generation. Summarization is critical for data-to-text generation because the amount of collected data may be massive. Examples of such applications include the summarization of intensive care unit data in the medical domain (Portet et al., 2009) and data from gas turbine sensors (Yu et al., 2007). Our approach is similar except that our input is text data in the form of conversations. We otherwise utilize a very similar architecture of *pattern recognition*, *pattern abstraction*, *pattern selection* and *summary generation*.

Kleinbauer et al. (2007) carry out topic-based meeting abstraction. Our systems differ in two major respects: their summarization process uses human gold-standard annotations of topic segments, topic labels and content items from the ontology, while our summarizer is fully automatic; secondly, the ontology they used is specific not just to meetings but to the AMI scenario meetings (Carletta et al., 2005), while our ontology applies to conversations in general, allowing our approach to be extended to emails, blogs, etc.

In this work we conduct a user study where participants use summaries to browse meeting transcripts. Some previous work has compared extracts and abstracts for the task of a decision audit (Murray et al., 2009), finding that human abstracts are a challenging gold-standard in terms of enabling participants to work quickly and correctly identify the relevant information. For that task, automatic extracts and the semi-automatic abstracts of Kleinbauer et al. (2007) were found to be competitive with one another in terms of user satisfaction and resultant task scores. Other research on comparing extracts and abstracts has found that an automatic abstractor outperforms a generic extractor in the domains of technical articles (Saggion and Lapalme, 2002) and evaluative reviews (Carenini and Cheung, 2008), and that human-written abstracts were rated best overall.

### 3 Interpretation - Ontology Mapping

Source document interpretation in our system relies on a general conversation ontology. The ontology is written in OWL/RDF and contains upper-level classes such as Participant, Entity, Utterance, and DialogueAct. When additional information is available about participant roles in a given domain, Participant subclasses such as ProjectManager can

be utilized. Object properties connect instances of ontology classes; for example, the following entry in the ontology states that the object property *hasSpeaker* has an instance of Utterance as its domain and an instance of Participant as its range.

```
<owl:ObjectProperty rdf:about="#hasSpeaker">
  <rdfs:range rdf:resource="#Participant"/>
  <rdfs:domain rdf:resource="#Utterance"/>
</owl:ObjectProperty>
```

The DialogueAct class has subclasses corresponding to a variety of sentence-level phenomena: decisions, actions, problems, positive-subjective sentences, negative-subjective sentences and general extractive sentences (important sentences that may not match the other categories). Utterance instances are connected to DialogueAct subclasses through an object property *hasDAType*. A single utterance may correspond to more than one DialogueAct; for example, it may represent both a positive-subjective sentence and a decision.

Our current definition of Entity instances is simple. The entities in a conversation are noun phrases with mid-range document frequency. This is similar to the definition of concept proposed by Xie et al. (2009), where n-grams are weighted by *tf.idf* scores, except that we use noun phrases rather than any n-grams because we want to refer to the entities in the generated text. We use mid-range document frequency instead of *idf* (Church and Gale, 1995), where the entities occur in between 10% and 90% of the documents in the collection. We do not currently attempt coreference resolution for entities; recent work has investigated coreference resolution for multi-party dialogues (Muller, 2007; Gupta et al., 2007), but the challenge of resolution on such noisy data is highlighted by low accuracy (e.g. F-measure of 21.21) compared with using well-formed text.

We map sentences to our ontology classes by building numerous supervised classifiers trained on labeled decision sentences, action sentences, etc. A general extractive classifier is also trained on sentences simply labeled as important. We give a specific example of the ontology mapping using the following excerpt from the AMI corpus, with entities italicized and resulting sentence classifications shown in bold:

- A: And you two are going to work together on a *prototype* using *modelling clay*. [**action**]
- A: You'll get *specific instructions* from your *personal coach*. [**action**]
- C: Cool. [**positive-subjective**]

- A: Um did we decide on a *chip*? [**decision**]
- A: Let's go with a *simple chip*. [**decision, positive-subjective**]

The ontology is populated by adding all of the sentence entities as instances of the Entity class, all of the participants as instances of the Participant class (or its subclasses such as ProjectManager when these are represented), and all of the utterances as instances of Utterance with their associated *hasDAType* properties indicating the utterance-level phenomena of interest. Here we show a sample Utterance instance:

```
<Utterance rdf:about="#ES2014a.B.dact.37">
<hasSpeaker rdf:resource="#IndustrialDesigner"/>
<hasDAType rdf:resource="#PositiveSubjective"/>
<begTime>456.58</begTime>
<endTime>458.832</endTime>
</Utterance>
```

### 3.1 Feature Set

The interpretation component as just described relies on supervised classifiers for the detection of items such as decisions, actions, and problems. This component uses general features that are applicable to any conversation domain. The first set of features we use for this ontology mapping are features relating to conversational structure. They include sentence length, sentence position in the conversation and in the current turn, pause-style features, lexical cohesion, centroid scores, and features that measure how terms cluster between conversation participants and conversation turns.

While these features have been found to work well for generic extractive summarization (Murray and Carenini, 2008), we use additional features for capturing the more specific sentence-level phenomena of this research. These include character trigrams, word bigrams, part-of-speech bigrams, word pairs, part-of-speech pairs, and varying instantiation n-grams, described in more detail in (Murray et al., 2010). After removing features that occur fewer than five times, we end up with 218,957 total features.

### 3.2 Message Generation

Rather than merely classifying individual sentences as decisions, action items, and so on, we also aim to detect larger patterns – or *messages* – within the meeting. For example, a given participant may repeatedly make positive comments about an entity throughout the meeting, or may give contrasting opinions of an entity. In order to determine which messages are essential for

summarizing meetings, three human judges conducted a detailed analysis of four development set meetings. They first independently examined previously-written human abstracts for the meetings to identify which messages were present in the summaries. In the second step, the judges met together to decide on a final message set. This resulted in a set of messages common to all the meetings and agreed upon by all the judges. The messages that our summarizer will automatically generate are defined as follows:

- *OpeningMessage* and *ClosingMessage*: Briefly describes opening/closing of the meeting
- *RepeatedPositiveMessage* and *RepeatedNegativeMessage*: Describes a participant making positive/negative statements about a given entity
- *ActionItemsMessage*: Indicates that a participant has action items relating to some entity
- *DecisionMessage*: Indicates that a participant was involved in a decision-making process regarding some entity
- *ProblemMessage*: Indicates that a participant repeatedly discussed problems or issues about some entity
- *GeneralDiscussionMessage*: Indicates that a participant repeatedly discussed a given entity

Message generation takes as input the ontology mapping described in the previous section, and outputs a set of messages for a particular meeting. This is done by identifying pairs of Participants and Entities that repeatedly co-occur with the various sentence-level predictions. For example, if the project manager repeatedly discusses the interface using utterances that are classified as positive-subjective, a *RepeatedPositiveMessage* is generated for that Participant-Entity pair. Messages are generated in a similar fashion for all other message types except for the opening and closing messages. These latter two messages are created simply by identifying which participants were most active in the introductory and concluding portions of the meeting and generating messages that describe that participant opening or closing the meeting.

Message types are defined within the OWL ontology, and the ontology is populated with message instances for each meeting. The following message describes the Marketing Expert making a decision concerning the television, and lists the relevant sentences contained by that decision message.

```
<DecisionMessage rdf:about="#dec9">
<messageSource rdf:resource="#MarketingExpert"/>
<messageTarget rdf:resource="#television"/>
<containsUtterance rdf:resource="#ES2014a.D.dact.55"/>
<containsUtterance rdf:resource="#ES2014a.D.dact.63"/>
</DecisionMessage>
```

## 4 Transformation - ILP Content Selection for Messages

Having detected all the messages for a given meeting conversation, we now turn to the task of transforming the source representation to a summary representation, which involves identifying the most informative messages for which we will generate text. We choose an integer linear programming (ILP) approach to message selection. ILP has previously been used for sentence selection in an extractive framework. Xie et al. (2009) used ILP to create a summary by maximizing a global objective function combining sentence and entity weights. Our method is similar except that we are selecting messages based on optimizing an objective function combining message and sentence weights:

$$\text{maximize } (1 - \lambda) * \sum_i w_i s_i + \lambda * \sum_j u_j m_j \quad (1)$$

$$\text{subject to } \sum_i l_i s_i < L \quad (2)$$

where  $w_i$  is the score for sentence  $i$ ,  $u_j$  is the score for message  $j$ ,  $s_i$  is a binary variable indicating whether sentence  $i$  is selected,  $m_j$  is a binary variable indicating whether message  $j$  is selected,  $l_i$  is the length of sentence  $i$  and  $L$  is the desired summary length. The  $\lambda$  term is used to balance sentence and message weights. Our sentence weight  $w_i$  is the sum of all the posterior probabilities for sentence  $i$  derived from the various sentence-level classifiers. In other words, sentences are weighted highly if they correspond to multiple object properties in the ontology. To continue the example from Section 3, the sentence *Let's go with the simple chip* will be highly weighted because it represents both a decision and a positive-subjective opinion. The message score  $u_j$  is the number of sentences contained by the message  $j$ . For instance, the DecisionMessage at the end of Section 3.2 contains two sentences. We can create a higher level of abstraction in our summaries if we select messages which contain numerous utterances. Similar to how sentences and concepts are combined in the previous ILP extraction approach (Xie et al., 2009; Gillick et al., 2009), messages and sentences are tied together by two additional constraints:

$$\sum_j m_j o_{ij} \geq s_i \quad \forall_i \quad (3)$$

$$m_j o_{ij} \leq s_i \quad \forall_{ij} \quad (4)$$

where  $o_{ij}$  is the occurrence of sentence  $i$  in message  $j$ . These constraints state that a sentence can only be selected if it occurs in a message that is selected, and that a message can only be selected if all of its sentences have also been selected.

For these initial experiments,  $\lambda$  is set to 0.5. The summary length  $L$  is set to 15% of the conversation word count. Note that this is a constraint on the length of the selected utterances; we additionally place a length constraint on the generated summary described in the following section. The reason for both types of length constraint is to avoid creating an abstract that is linked to a great many conversation utterances but is very brief and likely to be vague and uninformative.

## 5 Summary Generation

The generation component of our system follows the standard pipeline architecture (Reiter and Dale, 2000), comprised of a text planner, a micro-planner and a realizer. We describe each of these in turn.

### 5.1 Text Planning

The input to the document planner is an ontology which contains the selected messages from the content selection stage. We take a top-down, schema-based approach to document planning (Reiter and Dale, 2000). This method is effective for summaries with a canonical structure, as is the case with meetings. There are three high-level schemas invoked in order: opening messages, body messages, and closing messages. For the body of the summary, messages are retrieved from the ontology using SPARQL, an SQL-style query language for ontologies, and are clustered according to entities. Entities are temporally ordered according to their average timestamp in the meeting. In the overall document plan tree structure, the body plan is comprised of document sub-plans for each entity, and the document sub-plan for each entity is comprised of document sub-plans for each message type. The output of the document planner is a tree structure with messages as its leaves and document plans for its internal

nodes. Our text planner is implemented within the Jena semantic web programming framework<sup>1</sup>.

## 5.2 Microplanning

The microplanner takes the document plan as input and performs two operations: aggregation and generation of referring expressions.

### 5.2.1 Aggregation

There are several possibilities for aggregation in this domain, such as aggregating over participants, entities and message types. The analysis of our four development set meetings revealed that aggregation over meeting participants is quite common in human abstracts, so our system supports such aggregation. This involves combining messages that differ in participants but share a common entity and message type; for example, if there are two `RepeatedPositiveMessage` instances about the user interface, one with the project manager as the source and one with the industrial designer as the source, a single `RepeatedPositiveMessage` instance is created that contains two sources. We do not aggregate over entities for the sole reason that the text planner already clustered messages according to entity. The entity clustering is intended to give the summary a more coherent structure but has the effect of prohibiting aggregation over entities.

### 5.2.2 Referring Expressions

To reduce redundancy in our generated abstracts, we generate alternative referring expressions when a participant or an entity is mentioned multiple times in sequence. For participants, this means the generation of a personal pronoun. For entities, rather than referring repeatedly to, e.g., *the remote control*, we generate expressions such as *that issue* or *this matter*.

## 5.3 Realization

The text realizer takes the output of the microplanner and generates a textual summary of a meeting. This is accomplished by first associating elements of the ontology with linguistic annotations. For example, participants are associated with a noun phrase denoting their role, such as *the project manager*. Since entities were defined simply as noun phrases with mid-frequency IDF scores, an entity instance is associated with that noun phrase. Messages themselves are associated with verbs,

subject templates and object templates. For example, instances of `DecisionMessage` are associated with the verb *make*, have a subject template set to the noun phrase of the message source, and have an object template *[NP a decision PP [concerning \_\_\_\_\_]]* where the object of the prepositional phrase is the noun phrase associated with the message target.

To give a concrete example, consider the following decision message:

```
<DecisionMessage rdf:about="#dec9">
<rdf:type rdf:resource="#owl:Thing"/>
<hasVerb>make</hasVerb>
<hasCompl>a decision</hasCompl>
<messageSource rdf:resource="#MarketingExpert"/>
<messageSource rdf:resource="#ProjectManager"/>
<messageTarget rdf:resource="#television"/>
<containsUtterance rdf:resource="#ES2014a.D.dact.55"/>
<containsUtterance rdf:resource="#ES2014a.D.dact.63"/>
</DecisionMessage>
```

There are two message sources, `ProjectManager` and `MarketingExpert`, and one message target, `television`. The subjects of the message are set to be the noun phrases associated with the marketing expert and the project manager, while the object template is filled with the noun phrase *the television*. This message is realized as *The project manager and the marketing expert made a decision about the television*.

For our realizer we use `simpleNLG`<sup>2</sup>. We traverse the document plan output by the microplanner and generate a sentence for each message leaf. A new paragraph is created when both the message type and target of the current message are different than the message type and target for the previous message.

## 6 Task-Based User Study

We carried out a formative user study in order to inform this early work on automatic conversation abstraction. This task required participants to review meeting conversations within a short timeframe, having a summary at their disposal. We compared human abstracts and extracts with our automatically generated abstracts. The interpretation component and a preliminary version of the transformation component have already been tested in previous work (Murray et al., 2010). The sentence-level classifiers were found to perform well according to the area under the receiver operator characteristic (AUROC) metric, which evaluates the true-positive/false-positive ratio as the

<sup>1</sup>to be made publicly available upon publication

<sup>2</sup><http://www.csd.abdn.ac.uk/~ereiter/simplenlg/>

posterior threshold is varied, with scores ranging from 0.76 for subjective sentences to 0.92 for action item sentences. In the following, we focus on the formative evaluation of the complete system. We first describe the corpus we used, then the materials, participants and procedure. Finally we discuss the study results.

## 6.1 AMI Meeting Corpus

For our meeting summarization experiments, we use the *scenario* portion of the AMI corpus (Carletta et al., 2005), where groups of four participants take part in a series of four meetings and play roles within a fictitious company. There are 140 of these meetings in total. For the *summary annotation*, annotators wrote abstract summaries of each meeting and extracted sentences that best conveyed or supported the information in the abstracts. The human-authored abstracts each contain a general abstract summary and three subsections for “decisions,” “actions” and “problems” from the meeting. A many-to-many mapping between transcript sentences and sentences from the human abstract was obtained for each annotator. Approximately 13% of the total transcript sentences are ultimately labeled as extracted sentences. A sentence is considered a decision item if it is linked to the decision portion of the abstract, and action and problem sentences are derived similarly. We additionally use subjectivity and polarity annotations for the AMI corpus (Wilson, 2008).

## 6.2 Materials, Participants and Procedures

We selected five AMI meetings for this user study, with each stage of the four-stage AMI scenario represented. The meetings average approximately 500 sentences each. We included the following three types of summaries for each meeting: (EH) gold-standard *human extracts*, (AH) gold-standard *human abstracts* described in Section 6.1, and (AA) the *automatic abstracts* output by our abstractor. All three conditions feature manual transcriptions of the conversation. Each summary contains links to the sentences in the meeting transcript. For extracts, this is a one-to-one mapping. For the two abstract conditions, this can be a many-to-many mapping between abstract sentences and transcript sentences.

Participants were given instructions to browse each meeting in order to understand the gist of the meeting, taking no longer than 15 minutes per

meeting. They were asked to consider the scenario in which they were a company employee who wanted to quickly review a previous meeting by using a browsing interface designed for this task. Figure 1 shows the browsing interface for meeting IS1001d with an automatically generated abstract on the left-hand side and the transcript on the right. In the screenshot, the user has clicked the abstract sentence *The industrial designer made a decision on the cost* and has been linked to a transcript utterance, highlighted in yellow, which reads *Also for the cost, we should only put one battery in it*. Notice that this output is not entirely correct, as the decision pertained to the battery, which impacted the cost. This sentence was generated because the entity *cost* appeared in several decision sentences.

The time constraint meant that it was not feasible to simply read the entire transcript straight through. Participants were free to adopt whatever browsing strategy suited them, including skimming the transcript and using the summary as they saw fit. Upon finishing their review of each meeting, participants were asked to rate their level of agreement or disagreement on several Likert-style statements relating to the difficulty of the task and the usefulness of the summary. There were six statements to be evaluated on a 1-5 scale, with 1 indicating strong disagreement and 5 indicating strong agreement:

- Q1: I understood the overall content of the discussion.
- Q2: It required a lot of effort to review the meeting in the allotted time.
- Q3: The summary was coherent and readable.
- Q4: The information in the summary was relevant.
- Q5: The summary was useful for navigating the discussion.
- Q6: The summary was missing relevant information.

Participants were also asked if there was anything they would have liked to have seen in the summary, and whether they had any general comments on the summary.

We recruited 19 participants in total, with each receiving financial reimbursement for their participation. Each participant saw one summary per meeting and rated every summary condition during the experiment. We varied the order of the meetings and summary conditions. With 19 subjects, three summary conditions and six Likert statements, we collected a total of 342 user judgments. To ensure fair comparison between the three summary types, we limit summary length to

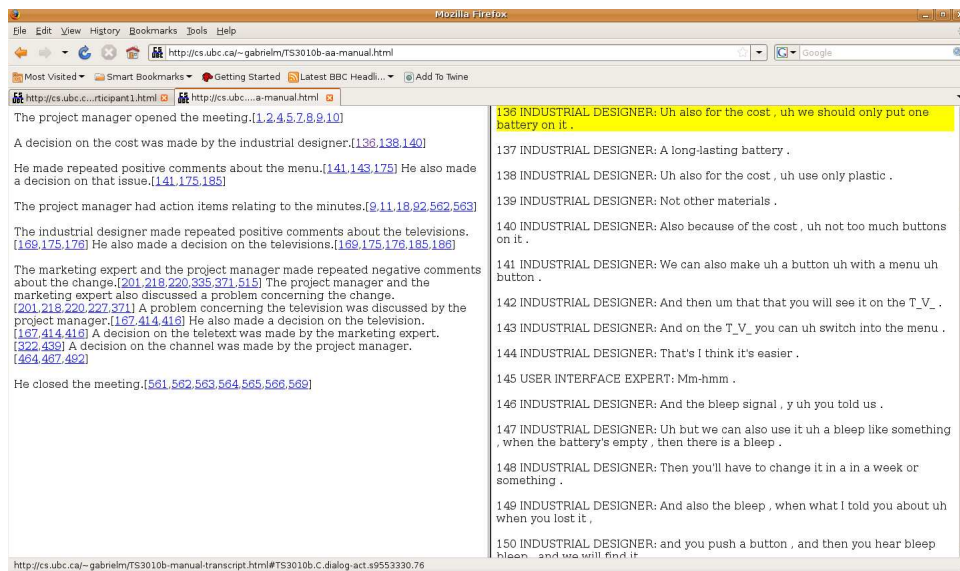


Figure 1: Summary Interface

be equal to the length of the human abstract for each meeting. This ranges from approximately 190 to 350 words per meeting summary.

### 6.2.1 Results and Discussion

Participants took approximately 12 minutes on average to review each meeting, slightly shorter than the maximum allotted fifteen minutes.

Figure 2 shows the average ratings for each summary condition on each Likert statement. For Q1, which concerns general comprehension of the meeting discussion, condition AH (human abstracts) is rated significantly higher than EH (human extracts) and AA (automatic abstracts) ( $p=0.0016$  and  $p=0.0119$  according to t-test, respectively). However, for the other statement that addresses the overall task, Q2, AA is rated best overall. Note that for Q2 a lower score is better. While there are no significant differences on this criterion, it is a compelling finding that automatic abstracts can greatly reduce the effort required for reviewing the meeting, at a level comparable to human abstracts.

Q3 concerns coherence and readability. Condition AH is significantly better than both EH and AA ( $p<0.0001$  and  $p=0.0321$ ). Our condition AA is also significantly better than the extractive condition EH ( $p=0.0196$ ). In the introduction we mentioned that a potential weakness of extractive summaries is that coherence and readability decrease when sentences are removed from their original contexts, and that extracts of noisy, unstructured source documents will tend to be noisy and un-

structured as well. These ratings confirm that extracts are not rated well on coherence and readability.

Q4 concerns the perceived relevance of the summary. Condition AH is again significantly better than EH and AA (both  $p<0.0001$ ). AA is rated substantially higher than EH on summary relevance, but not at a significant level.

Q5 is a key question because it directly addresses the issue of summary usability for such a task. Condition AH is significantly better than EH and AA (both  $p<0.0001$ ), but we also find that AA is significantly better than EH ( $p=0.0476$ ). Extracts have an average score of only 2.37 out of 5, compared with 3.21 and 4.63 for automatic and human abstracts, respectively. For quickly reviewing a meeting conversation, abstracts are much more useful than extracts.

Q6 indicates whether the summaries were missing any relevant information. As with Q2, a lower score is better. Condition AH is significantly better than EH and AA ( $p<0.0001$  and  $p=0.0179$ ), while AA is better than EH with marginal significance ( $p=0.0778$ ). This indicates that our automatic abstracts were better at containing all the relevant information than were human-selected extracts.

All participants gave written answers to the open-ended questions, yielding insights into the strengths and weaknesses of the different summary types. Regarding the automatic abstracts (AA), the most common criticisms were that the

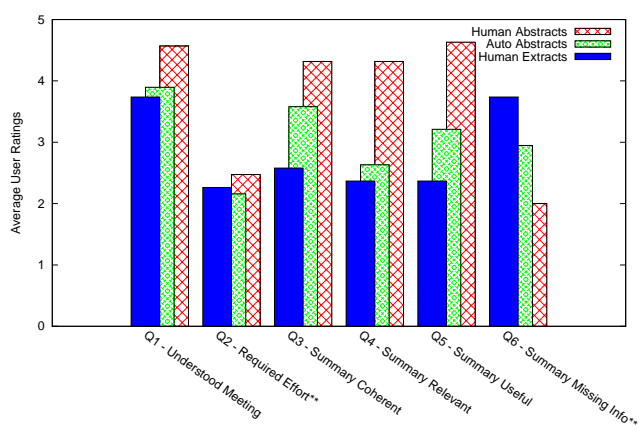


Figure 2: User Ratings (\*\* indicates lower score is better)

summaries are too vague (e.g. “more concrete would help”) and that the phrasing can be repetitive. There is a potential many-to-many mapping between abstract sentences and transcript sentences, and some participants felt that it was unnecessarily redundant to be linked to the same transcript sentence more than once (e.g. “quite a few repetitive citations”). Several participants felt that the sentences regarding positive-subjective and negative-subjective opinions were overstated and that the actual opinions were either more subtle or neutral. One participant wrote that these sentences constituted “a lot of bias in the summary.” On the positive side, several participants considered the links between abstract sentences and transcript sentences to be very helpful, e.g. “it really linked to the transcript well” and “I like how the summary has links connected to the transcript. Easier to follow-up on the meeting w/ the aid of the summary.” One participant particularly liked the subjectivity-oriented sentences: “Lifting some of the positive/negative from the discussion into the summary can mean the discussion does not even need to be included to get understanding.”

The written comments on the extractive condition (EH) were almost wholly negative. Many participants felt that the extracts did not even constitute a summary or that a cut-and-paste from the transcript does not make a sufficient summary (e.g. “The summary was not helpful @ all because it’s just cut from the transcript”, “All copy and paste not a summary”, “Not very clear summary - looked like the transcript”, and “No effort was made in the summary to put things into context”). Interestingly, several participants criti-

cized the extracts for not containing the most important sentences from the transcript despite these being human-selected extracts, demonstrating that a good summary is a subjective matter.

The comments on human abstracts (AH) were generally very positive, e.g. “easy to follow”, “it was good, clear”, and “I could’ve just read the summary and still understood the bulk of the meeting’s content.” The most frequent negative criticisms were that the abstract sentences sometimes contained too many links to the transcript (“massive amount of links look daunting”), and that the summaries were sometimes too vague (“perhaps some points from the discussion can be included, instead of just having topic outlines”, “[want] specific details”). It is interesting to observe that this latter criticism is shared between human abstracts and our automatic abstracts. When generalizing over the source document, details are sometimes sacrificed.

## 7 Conclusion

We have presented a system for automatically generating abstracts of meeting conversations. This summarizer relies on first mapping sentences to a conversation ontology representing phenomena such as decisions, action items and sentiment, then identifying message patterns that abstract over multiple sentences. We select the most informative messages through an ILP optimization approach, aggregate messages, and finally generate text describing all of the selected messages. A formative user study shows that, overall, our automatic abstractive summaries rate very well in comparison with human extracts, particularly regarding readability, coherence and usefulness. The automatic abstracts are also significantly better in terms of containing all of the relevant information (Q6), and it is impressive that an automatic abstractor substantially outperforms human-selected content on such a metric. In future work we aim to bridge the performance gap between automatic and human abstracts by identifying more specific messages and reducing redundancy in the sentence mapping. We plan to improve the NLG output by introducing more linguistic variety and better text structuring. We are also investigating the impact of ASR transcripts on abstracts and extracts, with encouraging early results.

**Acknowledgments** Thanks to Nicholas Fitzgerald for work on implementing the top-down planner.



## References

- R. Barzilay and K. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.
- G. Carenini and JCK Cheung. 2008. Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proc. of the 5th International Natural Generation Conference*.
- J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI meeting corpus: A pre-announcement. In *Proc. of MLMI 2005, Edinburgh, UK*, pages 28–39.
- K. Church and W. Gale. 1995. Inverse document frequency IDF: A measure of deviation from poisson. In *Proc. of the Third Workshop on Very Large Corpora*, pages 121–130.
- J. Clarke and M. Lapata. 2006. Constraint-based sentence compression: An integer programming approach. In *Proc. of COLING/ACL 2006*, pages 144–151.
- D. Gillick, K. Riedhammer, B. Favre, and D. Hakkani-Tür. 2009. A global optimization framework for meeting summarization. In *Proc. of ICASSP 2009, Taipei, Taiwan*.
- S. Gupta, J. Niekrasz, M. Purver, and D. Jurafsky. 2007. Resolving "You" in multi-party dialog. In *Proc. of SIGdial 2007, Antwerp, Belgium*.
- L. He, E. Sanocki, A. Gupta, and J. Grudin. 1999. Auto-summarization of audio-video presentations. In *Proc. of ACM MULTIMEDIA '99, Orlando, FL, USA*, pages 489–498.
- K. Spärck Jones. 1999. Automatic summarizing: Factors and directions. In I. Mani and M. Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–12. MITP.
- T. Kleinbauer, S. Becker, and T. Becker. 2007. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *Proc. of ENLG 2007, Dagstuhl, Germany*.
- K. Knight and D. Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proc. of AAAI 2000, Austin, Texas, USA*, pages 703–710.
- K. McKeown, J. Hirschberg, M. Galley, and S. Maskey. 2005. From text to speech summarization. In *Proc. of ICASSP 2005, Philadelphia, USA*, pages 997–1000.
- C. Muller. 2007. Resolving *It*, *This* and *That* in unrestricted multi-party dialog. In *Proc. of ACL 2007, Prague, Czech Republic*.
- G. Murray and G. Carenini. 2008. Summarizing spoken and written conversations. In *Proc. of EMNLP 2008, Honolulu, HI, USA*.
- G. Murray, T. Kleinbauer, P. Poller, S. Renals, T. Becker, and J. Kilgour. 2009. Extrinsic summarization evaluation: A decision audit task. *ACM Transactions on SLP*, 6(2).
- G. Murray, G. Carenini, and R. Ng. 2010. Interpretation and transformation for abstracting conversations. In *Proc. of NAACL 2010, Los Angeles, USA*.
- F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173:789–816.
- E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, GB.
- H. Saggion and G. Lapalme. 2002. Generating indicative-informative summaries with sumum. *Computational Linguistics*, 28(4):497–526.
- T. Wilson. 2008. Annotating subjective content in meetings. In *Proc. of LREC 2008, Marrakech, Morocco*.
- S. Xie, B. Favre, D. Hakkani-Tür, and Y. Liu. 2009. Leveraging sentence weights in a concept-based optimization framework for extractive meeting summarization. In *Proc. of Interspeech 2009, Brighton, England*.
- J. Yu, E. Reiter, J. Hunter, and C. Mellish. 2007. Choosing the content of textual summaries of large time-series data sets. *Journal of Natural Language Engineering*, 13:25–49.



# Charting the Potential of Description Logic for the Generation of Referring Expressions

Yuan Ren and Kees van Deemter and Jeff Z. Pan

Department of Computing Science  
University of Aberdeen  
Aberdeen, UK

## Abstract

The generation of referring expressions (GRE), an important subtask of Natural Language Generation (NLG) is to generate phrases that uniquely identify domain entities. Until recently, many GRE algorithms were developed using only simple formalisms, which were tailor made for the task. Following the fast development of ontology-based systems, reinterpretations of GRE in terms of description logic (DL) have recently started to be studied. However, the expressive power of these DL-based algorithms is still limited, not exceeding that of older GRE approaches. In this paper, we propose a DL-based approach to GRE that exploits the full power of OWL2. Unlike existing approaches, the potential of reasoning in GRE is explored.

## 1 GRE and KR: the story so far

Generation of Referring Expressions (GRE) is the subtask of Natural Language Generation (NLG) that focuses on identifying objects in natural language. For example, Fig.1 depicts the relations between several women, dogs and cats. In such a scenario, a GRE algorithm might identify  $d1$  as “the dog that loves a cat”, singling out  $d1$  from the five other objects in the domain. Reference has long been a key issue in theoretical linguistics and psycholinguistics, and GRE is a crucial component of almost every practical NLG system. In the years following seminal publications such as (Dale and Reiter, 1995), GRE has become one of the most intensively studied areas of NLG, with links to many other areas of Cognitive Science. After plan-based contributions (e.g., (Appelt, 1985)), recent work increasingly stresses the human-likeness of the expressions generated in simple situations, culminating in two evalua-

tion campaigns in which dozens of GRE algorithms were compared to human-generated expressions (Belz and Gatt, 2008; Gatt et al., 2009).

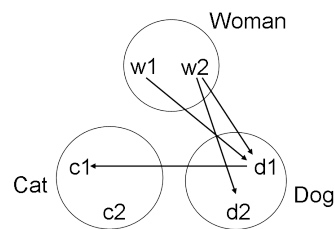


Figure 1: An example in which edges from women to dogs denote *feed* relations, from dogs to cats denote *love* relations.

Traditional GRE algorithms are usually based on very elementary, custom-made, forms of Knowledge Representation (KR), which allow little else than atomic facts (with negation of atomic facts left implicit), often using a simple  $\langle Attribute : Value \rangle$  format, e.g.  $\langle Type : Dog \rangle$ . This is justifiable as long as the properties expressed by these algorithms are simple one-place predicates (e.g., being a dog), but when logically more complex descriptions are involved, the potential advantages of “serious” KR become overwhelming. (This point will become clearer in later sections.) This realisation is now motivating a modest new line of research which stresses logical and computational issues, asking what properties a KR framework needs to make it suitable to generate all the referring expressions that people can produce (and to generate them in reasonable time). In this new line of work, which is proceeding in tandem with the more empirically oriented work mentioned above, issues of human-likeness are temporarily put on the backburner. These and other empirical issues will be brought to bear once it is better understood what types of KR system are best suitable for GRE, and what is the best way to pursue GRE in them.

A few proposals have started to combine GRE with KR. Following on from work based on labelled directed graphs (cf. (Krahmer et al., 2003)) – a well-understood mathematical formalism that offers no reasoning support – (Croitoru and van Deemter, 2007) analysed GRE as a *projection* problem in Conceptual Graphs. More recently, (Areces et al., 2008) analysed GRE as a problem in Description Logic (DL), a formalism which, like Conceptual Graphs, is specifically designed for representing and reasoning with potentially complex information. The idea is to produce a formula such as  $Dog \sqcap \exists love.Cat$  (the set of dogs intersected with the set of objects that love at least one cat); this is, of course, a successful reference if there exists *exactly one* dog who loves at least one cat. This approach forms the starting point for the present paper, which aims to show that when a principled, logic based approach is chosen, it becomes possible to refer to objects which no existing approach to GRE (including that of Areces et al.) has been able to refer to. To do this, we deviate substantially from these earlier approaches. For example, while Areces et al. use one finite interpretation for model checking, we consider arbitrary (possibly infinite) interpretations, hence reasoning support becomes necessary.

We shall follow many researchers in focussing on the semantic core of the GRE problem: we shall generate descriptions of semantic content, leaving the decision of what words to use for expressing this content (e.g., ‘the ancient dog’, or ‘the dog which is old’) to later stages in the NLG pipeline. Furthermore, we assume that all domain objects are equally salient (Krahmer and Theune, 2002). As explained above, we do not consider here the important matter of the naturalness or efficacy of the descriptions generated. We shall be content producing uniquely referring expressions whenever such expressions are possible, leaving the choice of the *optimal* referring expression in each given situation for later.

In what follows, we start by explaining how DL has been applied in GRE before (Sec. 2), pointing out the limitations of existing work. In Sec.3 we discuss which kinds of additional expressivity are required and how they can be achieved through modern DL. In Sec.4 we present a generic algorithm to compute these expressive REs. Sec.5 concludes the paper by comparing its aims and achievements with current practise in GRE.

## 2 DL for GRE

### 2.1 Description Logics

Description Logic (DLs) come in different flavours, based on decidable fragments of first-order logic. A DL-based KB represents the domain with descriptions of concepts, relations, and their instances. DLs underpin the Web Ontology Language (OWL), whose latest version, OWL2 (Motik et al., 2008), is based on DL *SR0IQ* (Horrocks et al., 2006).

An *SR0IQ* ontology  $\Sigma$  usually consists of a TBox  $\mathcal{T}$  and an ABox  $\mathcal{A}$ .  $\mathcal{T}$  contains a set of concept inclusion axioms of the form  $C \sqsubseteq D$ , relation inclusion axioms such as  $R \sqsubseteq S$  (the relation  $R$  is contained in the relation  $S$ ),  $R_1 \circ \dots \circ R_n \sqsubseteq S$ , and possibly more complex information, such as the fact that a particular relation is functional, or symmetric;  $\mathcal{A}$  contains axioms about individuals, e.g.  $a : C$  ( $a$  is an instance of  $C$ ),  $(a, b) : R$  ( $a$  has an  $R$  relation with  $b$ ).

Given a set of atomic concepts, the entire set of concepts expressible by *SR0IQ* is defined recursively. First, all atomic concepts are concepts. Furthermore, if  $C$  and  $D$  are concepts, then so are  $\top \mid \perp \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C \mid \leq nR.C \mid \geq nR.C \mid \exists R.Self \mid \{a_1, \dots, a_n\}$ , where  $\top$  is the top concept,  $\perp$  the bottom concept,  $n$  a non-negative integer number,  $\exists R.Self$  the self-restriction ((i.e., the set of those  $x$  such that  $(x, x) : R$  holds)),  $a_i$  individual names and  $R$  a relation which can either be an atomic relation or the inverse of another relation ( $R^-$ ). We call a set of individual names  $\{a_1, \dots, a_n\}$  a *nominal*, and use  $CN$ ,  $RN$  and  $IN$  to denote the set of atomic concept names, relation names and individual names, respectively.

An *interpretation*  $\mathcal{I}$  is a pair  $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$  where  $\Delta^{\mathcal{I}}$  is a non-empty set and  $\cdot^{\mathcal{I}}$  is a function that maps atomic concept  $A$  to  $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ , atomic role  $r$  to  $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$  and individual  $a$  to  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ . The interpretation of complex concepts and axioms can be defined inductively based on their semantics, e.g.  $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ , etc.

$\mathcal{I}$  is a *model* of  $\Sigma$ , written  $\mathcal{I} \models \Sigma$ , iff all the axioms in  $\Sigma$  are satisfied in  $\mathcal{I}$ . It should be noted that one  $\Sigma$  can have multiple models. For example when  $\mathcal{T} = \emptyset$ ,  $\mathcal{A} = \{a : A \sqcup B\}$ , there can be a model  $\mathcal{I}_1$  s.t.  $\Delta^{\mathcal{I}_1} = \{a\}$ ,  $a^{\mathcal{I}_1} = a$ ,  $A^{\mathcal{I}_1} = \{a\}$ ,  $B^{\mathcal{I}_1} = \emptyset$ , and another model  $\mathcal{I}_2$  s.t.  $\Delta^{\mathcal{I}_2} = \{a\}$ ,  $a^{\mathcal{I}_2} = a$ ,  $B^{\mathcal{I}_2} = \{a\}$ ,  $A^{\mathcal{I}_2} = \emptyset$ . In other words, the world is open. For details, see

(Horrocks et al., 2006).

The possibly multiple models indicate that an ontology is describing an open world. In GRE, researchers usually impose a closed world. From the DL point of view, people can (partially) close the ontology with a DBox  $\mathcal{D}$  (Seylan et al., 2009), which is syntactically similar to the ABox, except that  $\mathcal{D}$  contains only atomic formulas. Furthermore, every concept or relation appearing in  $\mathcal{D}$  is closed. Its extension is exactly defined by the contents of  $\mathcal{D}$ , i.e. if  $D \not\models a : A$  then  $a : \neg A$ , thus is the same in all the models. The concepts and relations not appearing in  $\mathcal{D}$  can still remain open. DL reasoning can be exploited to infer implicit information from ontologies. For example, given  $\mathcal{T} = \{Dog \sqsubseteq \exists feed^- . Woman\}$  (every dog is fed by some woman) and  $\mathcal{A} = \{d1 : Dog, w1 : Woman\}$ , we know that there must be some *Woman* who feeds *d1*. When the domain is closed as  $\mathcal{D} = \mathcal{A}$  we can further infer that this *Woman* is *w1* although there is no explicit relation between *w1* and *d1*. Note that the domain  $\Delta^{\mathcal{I}}$  in an interpretation of  $\mathcal{D}$  is not fixed, but it includes all the DBox individuals.

However, closing ontologies by means of the DBox can restrict the usage of implicit knowledge (from  $\mathcal{T}$ ). More precisely, the interpretations of the concepts and relations appearing in  $\mathcal{D}$  are fixed therefore no implicit knowledge can be inferred. To address this issue, we introduce the notion of NBox to support Negation as Failure (NAF): Under NAF, an ontology is a triple  $\mathcal{O} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$ , where  $\mathcal{T}$  is a TBox,  $\mathcal{A}$  an ABox and  $\mathcal{N}$  is a subset of *CN* or *RN*. We call  $\mathcal{N}$  an NBox. NAF requires that  $\mathcal{O}$  satisfy the following conditions:

1. Let  $x \in IN$  and  $A \in \mathcal{N} \cap CN$ . Then  $(\mathcal{T}, \mathcal{A}) \not\models x : A$  implies  $\mathcal{O} \models x : \neg A$ .
2. Let  $x, y \in IN$  and  $r \in \mathcal{N} \cap RN$ . Then  $(\mathcal{T}, \mathcal{A}) \not\models (x, y) : r$  implies  $\mathcal{O} \models (x, y) : \neg r$ .

Like the DBox approach, the NBox  $\mathcal{N}$  defines conditions in which “unknown” should be treated as “failure”. But, instead of hard-coding this, it specifies a vocabulary on which such treatment should be applied. Different from the DBox approach, inferences on this NAF vocabulary is still possible. An example of inferring implicit knowledge with NAF will be shown in later sections.

## 2.2 Background Assumptions

When applying DL to GRE, people usually impose the following assumptions.

- Individual names are not used in REs. For example, “the *Woman* who feeds *d1*” would be invalid, because *d1* is a name. Names are typically outlawed in GRE because, in many applications, many objects do not have names that readers/hearers would be familiar with.
- *Closed World Assumption (CWA)*: GRE researchers usually assume a closed world, without defining what this means. As explained above, DL allows different interpretations of the CWA. Our solution does not depend on a specific definition of CWA. In what follows, however, we use NAF to illustrate our idea. Furthermore, the domain is usually considered to be finite and consists of only individuals appearing in  $\mathcal{A}$ .
- *Unique Name Assumption (UNA)*: Different names denote different individuals. If, for example, *w1* and *w2* may potentially be the same woman, then we can not distinguish one from the other.

We follow these assumptions when discussing existing works and presenting our approach. In addition, we consider the entire KB, including  $\mathcal{A}$ ,  $\mathcal{T}$  and  $\mathcal{N}$ . It is also worth mentioning that, in the syntax of *SR**O**I**Q*, negation of relations are not allowed in concept expressions, e.g. you cannot compose a concept  $\exists \neg feed . Dog$ . However, if  $feed \in \mathcal{N}$ , then we can interpret  $(\neg feed)^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus feed^{\mathcal{I}}$ . In the rest of the paper, we use this as syntactic sugar.

## 2.3 Motivation: DL Reasoning and GRE

Every DL concept can be interpreted as a set. If the KB allows one to prove that this set is a singleton then the concept is a referring expression. It is this idea (Gardent and Striegnitz, 2007) that (Areces et al., 2008) explored. In doing so, they say little about the TBox, appearing to consider only the ABox, which contains only axioms about instances of atomic concepts and relations. For example, the domain in Fig.1 can be described as

KB1:  $\mathcal{T}_1 = \emptyset$ ,  $\mathcal{A}_1 = \{w1 : Woman, w2 : Woman, d1 : Dog, d2 : Dog, c1 : Cat, c2 : Cat, (w1, d1) : feed\}$ ,

$$\begin{aligned} (w2, d1) &: feed, (w2, d2) : feed, \\ (d1, c1) &: love \end{aligned}$$

Assuming that this represents a Closed World, Areces et al. propose an algorithm that is able to generate descriptions by partitioning the domain.<sup>1</sup> More precisely, the algorithm first finds out which objects are describable through increasingly large conjunctions of (possibly negated) atomic concepts, then tries to extend these conjunctions with complex concepts of the form  $(\neg)\exists R1.Concept$ , then with concepts of the form  $(\neg)\exists R2.(Concept \sqcap (\neg)\exists R1.Concept)$ , and so on. At each stage, only those concepts that have been acceptable through earlier stages are used. Consider, for instance, KB1 above. Regardless of what the intended referent is, the algorithm starts partitioning the domain stage by stage as follows. Each stage makes use of all previous stages. During stage (3), e.g., the object  $w2$  could only be identified because  $d2$  was identified in stage (2):

1.  $Dog = \{d1, d2\}$ ,  
 $\neg Dog \sqcap Woman = \{w1, w2\}$ ,  
 $\neg Dog \sqcap \neg Woman = \{c1, c2\}$ .
2.  $Dog \sqcap \exists love.(\neg Dog \sqcap \neg Woman) = \{d1\}$ ,  
 $Dog \sqcap \neg \exists love.(\neg Dog \sqcap \neg Woman) = \{d2\}$ .
3.  $(\neg Dog \sqcap Woman) \sqcap \exists feed.(Dog \sqcap \neg \exists love.(\neg Dog \sqcap \neg Woman)) = \{w2\}$ ,  
 $(\neg Dog \sqcap Woman) \sqcap \neg \exists feed.(Dog \sqcap \neg \exists love.(\neg Dog \sqcap \neg Woman)) = \{w1\}$ .

As before, we disregard the important question of the quality of the descriptions generated, other than whether they do or do not identify a given referent uniquely. Other aspects of quality depend in part on details, such as the question in which order atomic concepts are combined during phase (1), and analogously during later phases.

However this approach does not extend the expressive power of GRE. This is not because of some specific lapse on the part of the authors: it seems to have escaped the GRE community as a whole that relations can enter REs in a variety of alternative ways.

Furthermore, the above algorithm considers only the ABox, therefore background information

<sup>1</sup>Areces et al. (Areces et al., 2008) consider several DL fragments (e.g.,  $\mathcal{ALC}$  and  $\mathcal{EL}$ ). Which referring expressions are expressible, in their framework, depends on which DL fragment is chosen. Existential quantification, however, is the only quantifier that was used, and inverse relations are not considered.

will not be used. It follows that the domain always has a fixed single interpretation/model. Consequently the algorithm essentially uses model-checking, rather than full reasoning. We will show that when background information is involved, reasoning has to be taken into account. For example, suppose we extend Fig.1 with background (i.e., TBox) knowledge saying that *one should always feed any animal loved by an animal whom one is feeding*, while also adding a love edge (Fig.2) between  $d2$  and  $c2$ :

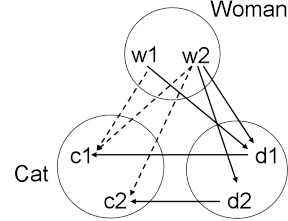


Figure 2: An extended example of Fig.1. Edges from women to cats denote *feed* relations. Dashed edges denote implicit relations.

If we close the domain with NAF, the ontology can be described as follows:

$$\begin{aligned} \text{KB2: } \mathcal{T}_2 &= \{feed \circ love \sqsubseteq feed\}, \\ \mathcal{A}_2 &= \mathcal{A}_1 \cup \{(d2, c2) : love\}, \mathcal{N}_2 = \\ &= \{Dog, Woman, feed, love\} \end{aligned}$$

The TBox axiom enables the inference of implicit facts: the facts  $(w1, c2) : feed$ ,  $(w2, c1) : feed$ , and  $(w2, c2) : feed$  can be inferred using DL reasoning under the above NBox  $\mathcal{N}_2$ . Axioms of this kind allow a much more natural, insightful and concise representation of information than would otherwise be possible.

Continuing to focus on the materialised KB2, we note another limitation of existing works: if only existential quantifiers are used then some objects are unidentifiable (i.e., it is not possible to distinguish them uniquely). These objects would become identifiable if other quantifiers and inverse relations were allowed. For example,

- The cat which is fed by *at least 2* women =  $Cat \sqcap \geq 2 feed^- . Woman = \{c1\}$ ,
- The woman feeding *only* those fed by *at least 2* women =  $Woman \sqcap \forall feed. \geq 2 . feed^- . Woman = \{w1\}$ ,
- The woman who feeds *all* the dogs =  $\{w2\}$ .

It thus raises the question: which quantifiers would it be natural to use in GRE, and how might DL realise them?

### 3 Beyond Existential Descriptions

In this section, we show how more expressive DLs can make objects referable that were previously unreferable. This will amount to a substantial reformulation which allows the approach based on DL reasoning to move well beyond other GRE algorithms in its expressive power.

#### 3.1 Expressing Quantifiers in OWL2

Because the proposal in (Areces et al., 2008) uses only existential quantification, it fails to identify *any* individual in Fig.2. Before filling this gap, we pause to ask what level of expressivity ought to be achieved. In doing so, we make use of a conceptual apparatus developed in an area of formal semantics and mathematical logic known as the theory of Generalized Quantifiers (GQ), where quantifiers other than *all* and *some* are studied (Mostowski, 1957). The most general format for REs that involves a relation  $R$  is, informally, the  $N1$  who  $R$   $Q$   $N2$ 's, where  $N1$  and  $N2$  denote sets,  $R$  denotes a relation, and  $Q$  a generalized quantifier. (Thus for example the women who feed SOME dogs.) An expression of this form is a unique identifying expression if it corresponds to exactly one domain element. Using a set-theoretic notation, this means that the following set has a cardinality of 1:

$$\{y \in N1 : Qx \in N2 \mid Ryx\}$$

where  $Q$  is a generalized quantifier. For example, if  $Q$  is the existential quantifier, while  $N1$  denotes the set of women,  $N2$  the set of dogs, and  $R$  the relation of feeding, then this says that the number of women who feed SOME dog is one. If  $Q$  is the quantifier *at least two*, then it says that the number of women who feed at least two dogs is one. It will be convenient to write the formula above in the standard GQ format where quantifiers are cast as relations between sets of domain objects  $A, B$ . Using the universal quantifier as an example, instead of writing  $\forall x \in A \mid x \in B$ , we write  $\forall(AB)$ . Thus, the formula above is written

$$\{y \in N1 : Q(N2\{z : Ryz\})\}.$$

Instantiating this as before, we get  $\{y \in Woman : \exists(Dog\{z : Feed\ yz\})\}$ , or “women who feed a dog”, where  $Q$  is  $\exists$ ,  $A = Dog$  and  $B = \{z : Feed\ yz\}$  for some  $y$ .

Mathematically characterising the class of *all* quantifiers that can be expressed in referring

expressions is a complex research programme to which we do not intend to contribute here, partly because this class includes quantifiers that are computationally problematic; for example, a quantifiers such as *most* (in the sense of more than 50%), which is not first-order expressible, as is well known.

To make transparent which quantifiers are expressible in the logic that we are using, let us think of quantifiers in terms of simple quantitative constraints on the sizes of the sets  $A \cap B$ ,  $A - B$ , and  $B - A$ , as is often done in GQ theory, asking what types of constraints can be expressed in referring expressions based on *SRQIQ*. The findings are summarised in Tab.1. OWL2 can express any of the following types of descriptions, plus disjunctions and conjunctions of anything it can express.

Table 1: Expressing GQ in DL

	$QAB$	$DL$
1	$\geq nN2\{z : Ryz\}$	$y : \geq nR.N2$
2	$\geq nN2\neg\{z : Ryz\}$	$y : \geq n\neg R.N2$
3	$\geq n\neg N2\{z : Ryz\}$	$y : \geq nR.\neg N2$
4	$\geq n\neg N2\neg\{z : Ryz\}$	$y : \geq n\neg R.\neg N2$
5	$\leq nN2\{z : Ryz\}$	$y : \leq nR.N2$
6	$\leq nN2\neg\{z : Ryz\}$	$y : \leq n\neg R.N2$
7	$\leq n\neg N2\{z : Ryz\}$	$y : \leq nR.\neg N2$
8	$\leq n\neg N2\neg\{z : Ryz\}$	$y : \leq n\neg R.\neg N2$

When  $n = 1$ , for example, type 1 becomes  $\exists R.N2$ , i.e. the *existential* quantifier. When  $n = 0$  type 7 becomes  $\forall R.N2$ , i.e. the quantifier *only*. When  $n = 0$  type 6 becomes  $\forall\neg R.\neg N2$ , i.e. the quantifier *all*. In types 2, 4, 6 and 8, negation of a relation is used. This is not directly supported in *SRQIQ* but, as we indicated earlier, given  $R \in \mathcal{N}$ ,  $\neg R$  can be used in concepts.

Together, this allows the expression of a description such as “women who feed at least one but at most 7 dogs”, by conjoining type 1 (with  $n = 1$ ) with type 5 (with  $n = 7$ ). Using negation, it can say “women who do not feed all dogs and who feed at least one non-dog” ( $Woman \sqcap \neg\forall\neg Feed.\neg Dog \sqcap \exists Feed.\neg Dog$ ). In addition to Tab.1, *SRQIQ* can even represent reflexive relation such as “the dog who loves itself” by  $Dog \sqcap \exists love.Self$ , which was regarded infeasible in (Gardent and Striegnitz, 2007).

Comparing the quantifiers that become expressible through OWL2's apparatus with classes of quantifiers studied in the theory of GQ, it is clear that OWL2 is highly expressive: it does not only

include quantifiers expressible in the binary tree of numbers, e.g. (van Benthem, 1986) – which is generally regarded as highly general – but much else besides. Even wider classes of referring expressions can certainly be conceived, but these are not likely to have overwhelming practical utility in today’s NLG applications.

#### 4 Generating *SROIQ*-enabled REs

In this section, we present an algorithm that computes the descriptions discussed in sect.3. A GRE algorithm should have the following behaviour: if an entity is distinguishable from all the others, the algorithm should find a unique description; otherwise, the algorithm should say there exists no unique description. In this paper, we follow Arces et al.’s strategy of generating REs for all objects simultaneously, but we apply it to a much larger search space, because many more constructs are taken into account.

##### 4.1 GROWL: an algorithm for Generating Referring expressions using OWL-2.

In this section we show how the ideas of previous sections can be implemented. To do this, we sketch an algorithm scheme called GROWL. GROWL applies a generate-and-test strategy that composes increasingly complicated descriptions and uses DL reasoning to test whether a description denotes a singleton w.r.t. the KB. To avoid considering unnecessarily complicated descriptions, the algorithm makes use of the (syntactic) depth of a description, defined as follows:

**Definition 1 (Depth)** Given a description  $d$ , its depth  $|d|$  is calculated as follows:

1.  $|d| = 1$  for  $d := \top \mid \perp \mid A \mid \neg A$ , where  $A$  is atomic.
2.  $|d \sqcap d'| = |d \sqcup d'| = \max(|d|, |d'|) + 1$ .
3.  $|\exists r.d| = |\forall r.d| = |\leq nr.d| = |\geq nr.d| = | = nr.d| = |d| + 1$ .

Different descriptions can mean the same of course, e.g.  $\neg\forall R.A \equiv \exists R.\neg A$ . We do not know which syntactic variant should be used but focus, for simplicity, on generating their unique *negated normal form* (NNF). The NNF of a formula  $\phi$  can be obtained by pushing all the  $\neg$  inward until only before atomic concepts (including  $\top$  and  $\perp$ ), atomic relations, nominals or self restrictions

(e.g.  $\exists r.Self$ ). Without loss of generality, in what follows we assume all the formulas are in their NNF. To avoid confusion, the NNF of negation of a formula  $\phi$  is denoted by  $\sim\phi$  instead of  $\neg\phi$ . For example  $\sim(A \sqcup B) = \neg A \sqcap \neg B$  if  $A$  and  $B$  are atomic. Obviously,  $\sim(\sim A) = A$ ,  $\sim(\sim R) = R$ ,  $(R^-)^- = R$ , and  $(\sim R)^- = \sim R^-$ . The use of NNF substantially reduces the redundancies generated by the algorithm. For example, we won’t generate both  $\neg\forall R.A$  and  $\exists R.\neg A$  but only the later.

Given an ontology  $\Sigma$ , we initialise GROWL with the following sets:

1. The relation name set  $RN$  is the minimal set satisfying:
  - if  $R$  is an atomic relation in  $\Sigma$ , then  $R \in RN$ ;
  - if  $R \in RN$ , then  $\sim R \in RN$ ;
  - if  $R \in RN$ , then  $R^- \in RN$ ;
2. The concept name set  $CN$  is the minimal set satisfying:
  - $\top \in CN$ ;
  - if  $A$  is an atomic concept in  $\Sigma$ , then  $A \in CN$ ;
  - if  $R \in RN$ , then  $\exists R.Self \in CN$ ;
  - if  $A \in CN$ , then  $\sim A \in CN$ ;
3. The natural number set  $N$  contains  $1, 2, \dots, n$  where  $n$  is the number of individuals in  $\Sigma$ .
4. The construct set  $S$  contains all the constructs supported by a particular language. For *SROIQ*,  $S = \{\neg, \sqcap, \sqcup, \exists, \forall, \leq, \geq, =\}$ . We assume here that nominals are disallowed (cf. sect.2).

---

##### Algorithm GROWL:

*Construct – description*( $\Sigma, CN, RN, N, S$ )

**INPUT:**  $\Sigma, CN, RN, N, S$

**OUTPUT:** Description Queue  $D$

- 1:  $D := \emptyset$
- 2: **for**  $e \in CN$  **do**
- 3:    $D := Add(D, e)$
- 4: **for**  $d = fetch(D)$  **do**
- 5:   **for each**  $s \in S$  **do**
- 6:     **if**  $s = \sqcap$  or  $s = \sqcup$  **then**
- 7:       **for each**  $d' \in D$  **do**
- 8:          $D := Add(D, d \ s \ d')$
- 9:     **if**  $s = \exists$  or  $s = \forall$  **then**



```

10:   for each  $r \in RN$  do
11:      $D := Add(D, s\ r.d)$ 
12:   if  $s = \leq$  or  $s = \geq$  or  $s\ is =$  then
13:     for each  $r \in RN$ , each  $k \in N$  do
14:        $D := Add(D, s\ k\ r.d)$ 
15: return  $D$ 

```

**Algorithm ADD:**  $Add(D, e)$

**INPUT:**  $D, e$

**OUTPUT:** (Extended) Description Queue  $D$

```

1: for  $d \in D$  do
2:   if  $|d| < |e|$  and  $d \sqsubseteq_{\Sigma} e$  then
3:     return  $D$ 
4:   else if  $|d| = |e|$  and  $d \sqsubseteq_{\Sigma} e$  and  $e \sqcap \neg d$  is
      satisfiable then
5:     return  $D$ 
6: if  $e$  is satisfiable in  $\Sigma$  then
7:    $D := D \cup \{e\}$ 
8: return  $D$ 

```

GROWL takes an ontology  $\Sigma$  as its input and outputs a queue  $D$  of descriptions by adding increasingly complex concepts  $e$  to  $D$ , using the function  $Add(D, e)$ , which is implemented as the algorithm ADD. Because of the centrality of ADD we start by explaining how this function works.

In the simple algorithm we are proposing in this paper – which represents only one amongst many possibilities – addition is governed by the heuristic that *more complex descriptions should have smaller extensions*. To this end, a candidate description  $e$  is compared with each existing description  $d \in D$ . Step 2 ensures that if there exists a simpler description  $d$  ( $|d| < |e|$ ) whose extension is no larger than  $e$  ( $d \sqsubseteq_{\Sigma} e$ ), then  $e$  is not added into  $D$  (because the role of  $e$  can be taken by the simpler description  $d$ ). Similarly, step 4 ensures that if there exists  $d$  with same depth ( $|d| = |e|$ ) but smaller extension ( $d \sqsubseteq_{\Sigma} e$  and  $e \sqcap \neg d$  is satisfiable), then  $e$  should not be added into  $D$ . The subsumption checking in Step 2 and 4, and the instance retrieval in Step 6, must be realised by DL reasoning, in which TBox, ABox and NBox must all be taken into account. ADD guarantees that when the complexity of descriptions increases, their extensions are getting smaller.

We now turn to the main algorithm, GROWL. In Step 1 of this algorithm,  $D$  is initialised to  $\emptyset$ . Steps 2 to 3 add all satisfiable elements of  $CN$  to  $D$ . From Steps 4 to 14, we recursively “process” ele-

ments of  $D$  one by one, by which we mean that the constructors in  $S$  are employed to combine these elements with other elements of  $D$  (e.g., an element is intersected with all other elements, and so on). We use  $fetch(D)$  to retrieve the first unprocessed element of  $D$ . New elements are added to the end of  $D$ . Thus  $D$  is a first-come-first-served queue (note that processed elements are not removed from  $D$ ).

To see in more detail how elements of  $D$  are processed, consider Steps 5-14 once again. For each element  $d$  of  $D$ , Step 5 uses a construct  $s$  to extend it:

1. If  $s$  is  $\sqcap$  or  $\sqcup$ , in Step 7 and 8, we extend  $d$  with each element of  $D$  and add new descriptions to  $D$ .
2. If  $s$  is  $\exists$  or  $\forall$ , in Step 10 and 11, we extend  $d$  with all relations of  $RN$  and add new descriptions to  $D$ . In Areces et al.’s work,  $\forall$  is also available when using  $\neg$  and  $\exists$  together, however due to their algorithm they can never generate descriptions like  $\forall r.A$ .
3. If  $s$  is  $\leq$ ,  $\geq$  or  $=$ , in Step 13 and 14, we extend  $d$  with all relations in  $RN$  and all numbers in  $N$ , and add new descriptions to  $D$ .

Because the  $=$  construct can be equivalently substituted by the combination of  $\leq$ ,  $\geq$  and  $\sqcap$  constructs ( $=\ kr.d$  is semantically equivalent to  $\geq\ kr.d \sqcap \leq\ kr.d$ ), it is a modelling choice to use either  $\leq$ ,  $\geq$ , or only  $=$ , or all of them. In this algorithm we use them all.

Because we compute only the NNF and we disallow the use of individual identifiers, negation  $\neg$  appears only in front of atomic concept names. For this reason, processing does not consider  $s = \neg$ . Note that GROWL leaves some important issues open. In particular, the order in which constructs, relations, integers and conjuncts/disjuncts are chosen is left unspecified. Note that  $D, RN, N, S$  are all assumed to be finite, hence Steps 5 to 14 terminate for a given  $d \in D$ . Because Steps 5 to 14 generate descriptions whose depth increases with one constructor at a time, there are finitely many  $d \in D$  such that  $|d| = n$  (for a given  $n$ ).

GROWL extends the algorithm presented by Areces et al. The example in Fig.2 shows that many referring expressions generated by our algorithm cannot be generated by our predecessors; in

fact, some objects that are not referable for them are referable by GROWL. For example, if we apply the algorithm to the KB in Fig.2, a possible solution is as follows:

1.  $\{w1\} = Woman \sqcap \exists \neg feed.Cat$ , the woman that does not feed all cats.
2.  $\{w2\} = \leq 0 \neg feed.Cat$ , the woman that feeds all cats.
3.  $\{d1\} = Dog \sqcap \leq 0 \neg feed^- .Woman$ , the dog that is fed by all women.
4.  $\{d2\} = Dog \sqcap \exists \neg feed^- .Woman$ , the dog that is not fed by all women.
5.  $\{c1\} = Cat \sqcap \leq 0 \neg feed^- .Woman$ , the cat that is fed by all women.
6.  $\{c2\} = Cat \sqcap \exists \neg feed^- .Woman$ , the cat that is not fed by all women.

It is worth reiterating here that our algorithm focusses on finding uniquely referring expressions, leaving aside which of all the possible ways in which an object can be referred to is “best”. For this reason, empirical validation of our algorithm – a very sizable enterprise in itself, which should probably be based on descriptions elicited by human speakers – is not yet in order.

## 4.2 Discussion

Let us revisit the basic assumptions of Sec.2.2, to see what can be achieved if they are abandoned.

1. In natural language, people do using names, e.g. “the husband of Marie Curie”. To allow REs of this kind, we can extend our Algorithm A-1 by including singleton classes such as  $\{Maria\_Curie\}$  in  $CN$ .
2. Traditional GRE approaches have always assumed a single model with complete knowledge. Without this assumption, our approach can still find interesting REs. For example, if a man’s nationality is unknown, but he is known to be the Chinese or Japanese, we can refer to him/her as  $Chinese \sqcup Japanese$ . However, models should be finite to guarantee that  $N$  is finite.
3. Individuals with multiple names. DL imposes the UNA by explicitly asserting the

inequality of each two individuals. Without UNA, reasoning can still infer some results, e.g.  $\{Woman \sqcap Man \sqsubseteq \perp, David : Man, May : Woman\} \models David \neq May$ . Thus we can refer to David as “the man” if the domain is closed.

## 5 Widening the remit of GRE

This paper has shown some of the benefits that arise when the power of KR is brought to bear on an important problem in NLG, namely the generation of referring expressions (GRE). We have done this by using DL as a representation and reasoning formalism, extending previous work in GRE in two ways. First, we have extended GRE by allowing the generation of REs that involve quantifiers other than  $\exists$ . By relating our algorithm to the theory of Generalised Quantifiers, we were able to formally characterise the set of quantifiers supported by our algorithm, making exact how much expressive power we have gained. Secondly, we have demonstrated the benefits of *implicit* knowledge through inferences that exploit TBox-information, thereby allowing facts to be represented more efficiently and elegantly, and allowing GRE to tap into kinds of generic (as opposed to atomic) knowledge that it had so far left aside, except for hints in (Gardent and Striegnitz, 2007) and in (Croitoru and van Deemter, 2007). Thirdly, we have allowed GRE to utilise incomplete knowledge, as when we refer to someone as “the man of Japanese or Chinese nationality”.

Current work on reference is overwhelmingly characterised by an emphasis on empirical accuracy, often focussing on very simple referring expressions, which are constituted by conjunctions of 1-place relations (as in “the grey poodle”), and asking which of these conjunctions are most likely to be used by human speakers (or which of these would be most useful to a human hearer). The present work stresses different concerns: we have focussed on questions of expressive power, focussing on relatively complex descriptions, asking what referring expressions are possible when relations between domain objects are used. We believe that, at the present stage of work in GRE, it is of crucial importance to gain insight into questions of this kind, since this will tell us what types of reference are possible in principle. Once such insight, we hope to explore how the newly gained expressive power can be put to practical use.

## References

- Douglas Appelt. 1985. *Planning English Sentences*. Cambridge University Press, Cambridge, UK.
- Carlos Areces, Alexander Koller, and Kristina Striegnitz. 2008. Referring expressions as formulas of description logic. In *Proceedings of the 5th INLG*, Salt Fork, Ohio.
- Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 197–200.
- Madalina Croitoru and Kees van Deemter. 2007. A conceptual graph approach to the generation of referring expressions. In *Proceedings of the 20th IJCAI*.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *CoRR*, cmp-lg/9504020.
- Claire Gardent and Kristina Striegnitz. 2007. Generating bridging definite descriptions. *Computing Meaning*, 3:369–396.
- Albert Gatt, Anja Belz, and Eric Kow. 2009. The TUNA-REG Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th ENLG (ENLG 2009)*, pages 174–182, Athens, Greece, March. Association for Computational Linguistics.
- Ian Horrocks, Oliver Kutz, and Ulrike Sattler. 2006. The Even More Irresistible SROIQ. In *KR 2006*.
- Emiel Krahmer and Mariet Theune. 2002. Efficient context-sensitive generation of descriptions in context. *Information Sharing: Givenness and Newness in Language*, pages 223–264.
- Emiel Krahmer, Sebastiaan van Erk, and Andr Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- A Mostowski. 1957. On a generalization of quantifiers. *Fund. Math.*, 44:235–273.
- Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. 2008. Owl 2 web ontology language: Profiles. W3c working draft, W3C, October.
- Inanç Seylan, Enrico Franconi, and Jos de Bruijn. 2009. Effective query rewriting with ontologies over dboxes. In *IJCAI 2009*.
- Johan van Benthem. 1986. *Essays in Logical Semantics*. Reidel.



# Complex Lexico-Syntactic Reformulation of Sentences using Typed Dependency Representations

Advaith Siddharthan

Department of Computing Science

University of Aberdeen

advait@abdn.ac.uk

## Abstract

We present a framework for reformulating sentences by applying transfer rules on a typed dependency representation. We specify a list of operations that the framework needs to support and argue that typed dependency structures are currently the most suitable formalism for complex lexico-syntactic paraphrasing. We demonstrate our approach by reformulating sentences expressing the discourse relation of *causation* using four lexico-syntactic discourse markers – “cause” as a verb and as a noun, “because” as a conjunction and “because of” as a preposition.

## 1 Introduction

There are many reasons why a writer might want to choose one formulation of a discourse relation over another; for example, maintaining thread of discourse, avoiding shifts in focus and issues of salience and end weight. There are also reasons to use different formulations for different audiences; for example, to account for differences in reading skills and domain knowledge. In recent work, Siddharthan and Katsos (2010) demonstrated through psycholinguistic experiments that domain experts and lay readers show significant differences in which formulations of *causation* they find acceptable. They further showed that the most appropriate formulation depends both on the domain expertise of the user and the propositional content of the sentence, and that these preferences can be learnt in a supervised machine learning framework. That work, as does much of the related comprehension and literacy literature, used manually reformulated sentences. In this paper, we present an approach to automate such complex reformulation. We consider the four lexico-syntactic discourse markers for *causation* studied by Siddharthan and Katsos (2010); consider 1a.–d. below (from their corpus, but simplified to aid presentation):

- (1) a. An incendiary device **caused** the explosion. [A-CAUSE-B]  
 b. The explosion occurred **because of** an incendiary device. [B-BECAUSEOF-A]  
 c. The explosion occurred **because** there was an incendiary device. [B-BECAUSE-A]  
 d. The **cause** of the explosion was an incendiary device. [CAUSEOF-B-A]

These differ in terms of the lexico-syntactic properties of the discourse marker (shown in bold font). Indeed the discourse markers here are verbs, prepositions, conjunctions and nouns. As a consequence, the propositional content is expressed either as a clause or a noun phrase (“*The explosion occurred*” vs “*the explosion*”, etc.). Additionally, the order of presentation of propositional content can be varied to give four more lexico-syntactic paraphrases:

- (1) e. The explosion **was caused by** an incendiary device. [B-CAUSEBY-A]  
 f. **Because of** an incendiary device, the explosion occurred. [BECAUSEOF-A-B]  
 g. **Because** there was an incendiary device, the explosion occurred. [BECAUSE-A-B]  
 h. An incendiary device was the **cause** of the explosion. [A-CAUSEOF-B]

It is clear that some formulations of a given propositional content can be more felicitous than others; for example, 1e. seems preferable to 1g. However, for different propositional content, other formulations might be more felicitous. While discourse level choices based on information ordering play a role in choosing a formulation, Siddharthan and Katsos (2010) demonstrate that some de-contextualised information orderings within a sentence are deemed unacceptable by some categories of readers. This has implications for text regeneration tasks that try to reformulate texts for different audiences; for instance, simplifying language for low reading ages or summarising technical writing for lay readers. In short, considerations of discourse coherence should not introduce sentence-level unacceptability in regenerated text.

We focus on causal relations for many reasons.

For the purpose of this paper, our main reason is that the 8 formulations selected are different information orderings of 4 different lexico-syntactic constructs. Thus, we explore a broad range of constructions and are confident that the framework we develop covers the range of operations required for text regeneration in general. Of less relevance to this paper, but equally important to our broad goals of reformulating technical writing for lay readers, causal relations are pervasive in science writing and are integral to how humans conceptualise the world. We have a particular interest in scientific writing – reformulating such texts for lay audiences is a highly relevant task today and many news agencies perform this service; e.g., Reuters Health summarises medical literature for lay audiences and BBC online has a Science/Nature section that reports on science. These services rely either on press releases by scientists and universities or on specialist scientific reporters, thus limiting coverage of a growing volume of scientific literature in a digital economy.

In Section 2, we relate our research to the existing linguistic and computational literature. Then in Section 3, we compare three different linguistic representations with respect to their suitability for lexico-syntactic reformulation. We found typed dependency structures to be the most promising and present an evaluation in Section 4.

## 2 Related Work

### 2.1 Discourse Connectives and Comprehension

Previous work has shown that when texts have been manually rewritten to make the language more accessible (L’Allier, 1980), or to make the content more transparent (Beck et al., 1991), students’ reading comprehension shows significant improvements. An example of a revision choice that might be applied differentially depending on the literacy skills of the reader involves connectives such as *because*. Connectives that permit pre-posed adverbial clauses have been found to be difficult for third to fifth grade readers, even when the order of mention coincides with the causal (and temporal) order (Anderson and Davison, 1988); this experimental result is consistent with the observed order of emergence of connectives in children’s narratives (Levy, 2003).

Thus the b) version of the following example would be preferred for children who can grasp causation, but who have not yet become comfortable with alternative clause orders (example from Anderson and Davison (1988), p. 35):

- (2) a. Because Mexico allowed slavery, many Americans and their slaves moved to Mexico during that time.
- b. Many Americans and their slaves moved to Mexico during that time, because Mexico allowed slavery.

Such studies show that comprehension can be improved by reformulating text for readers with low reading skills (Linderholm et al., 2000; Beck et al., 1991) and for readers with low levels of domain expertise (Noordman and Vonk, 1992). Further, specific information orderings were found to be facilitatory by Anderson and Davison (1988). All these studies suggest that the automatic lexico-syntactic reformulation of causation can benefit various categories of readers.

### 2.2 Connectives and Text (Re)Generation

Much of the work regarding (re)generation of text based on discourse connectives aims to simplify text in certain ways, to make it more accessible to particular classes of readers. The PSET project (Carroll et al., 1998) considered simplifying news reports for aphasics. The PSET project focused mainly on lexical simplification (replacing difficult words with easier ones), but there has been work on syntactic simplification and, in particular, the way syntactic rewrites interact with discourse structure and text cohesion (Siddharthan, 2006). These were restricted to string substitution and sentence splitting based on pattern matching over chunked text. Our work aims to extend these strands of research by allowing for more sophisticated insertion, deletion and substitution operations that can involve substantial reorganisation and modification of content within a sentence.

Elsewhere, there has been interest in *paraphrasing*, including the replacement of words (especially verbs) with their dictionary definitions (Kaji et al., 2002) and the replacement of idiomatic or otherwise troublesome expressions with simpler ones. The emphasis has been on automatically learning paraphrases from comparable or aligned corpora (Barzilay and Lee, 2003; Ibrahim et al., 2003). The text simplification and paraphrasing literature does not address paraphrasing that requires syntactic alterations such as those in Example 1 or the question of appropriateness of different formulations of a discourse relation.

Some natural language generation systems incorporate results from psycholinguistic studies to make principled choices between alternative formulations. For example, SkillSum (Williams and Reiter, 2008) and ICONOCLAST (Power et al.,

2003) are two contemporary generation systems that allow for specifying aspects of style such as choice of discourse marker, clause order, repetition and sentence and paragraph lengths in the form of constraints that can be optimised. However, to date, these systems do not consider syntactic reformulations of the type we are interested in. Our research is directly relevant to such generation systems as it can help such systems make decisions in a principled manner.

Williams et al. (2003) examined the impact of discourse level choices on readability in the domain of reporting the results of literacy assessment tests, using the results of the test to control both the content and the realisation of the generated report. Our research aims to facilitate the transfer of such user-driven generation research to text regeneration areas.

### 2.3 Sentence Compression

Sentence compression is a research area that aims to shorten sentences for the purpose of summarising the main content. There are similarities between our interest in reformulation and existing work in sentence compression. Sentence compression has usually been addressed in a generative framework, where transformation rules are learnt from parsed corpora of sentences aligned with manually compressed versions. The compression rules learnt are therefore tree-tree transformations (Knight and Marcu, 2000; Galley and McKeown, 2007; Riezler et al., 2003) of some variety. These approaches focus on *deletion* operations, mostly performed low down in the parse tree to remove modifiers. Further they make assumptions about isomorphism between the aligned tree, which means they cannot be readily applied to more complex reformulation operations such as *insertion* and *reordering* that are essential to perform reformulations such as those in Example 1. Cohn and Lapata (2009) provide an approach based on Synchronous Tree Substitution Grammar (STSG) that in principle can handle the range of reformulation operations. However, given their focus on sentence compression, they restricted themselves to local transformations near the bottom of the parse tree. In this paper, we explore whether this framework could prove useful to more involved reformulation tasks. Our experience (see Section 3.2) suggests that parse trees are the wrong representation for learning complex transformation rules and that dependency structures are more suited for complex lexico-syntactic reformulation.

## 3 Regeneration using Transfer Rules

We experimented with three representations – phrasal parse trees, typed dependencies and Minimal Recursion Semantics (MRS). In this section, we first describe our data, and then report our experience with performing text reformulation using these representations.

### 3.1 Data

We use the corpus described in Siddharthan and Katsos (2010). This corpus contains examples of complex lexico-syntactic reformulations such as those in Example 1a–f; each example consists of 8 formulations, 7 of which are manual reformulations. The corpus contains 144 such examples from three genres, giving 1152 sentences in total. The manual reformulation is formulaic and Example 1 is indicative of the process. To make a clause out of a noun phrase, either the copula or the verb “occur” is introduced, based on a subjective judgement of whether this is an event or a continuous phenomenon. Conversely, to create a noun phrase from a clause, a possessive and gerund are used; for example (from Siddharthan and Katsos (2010)):

- (3) a. Irwin had triumphed because he was so good a man.  
 b. The cause of Irwin’s having triumphed was his being so good a man.

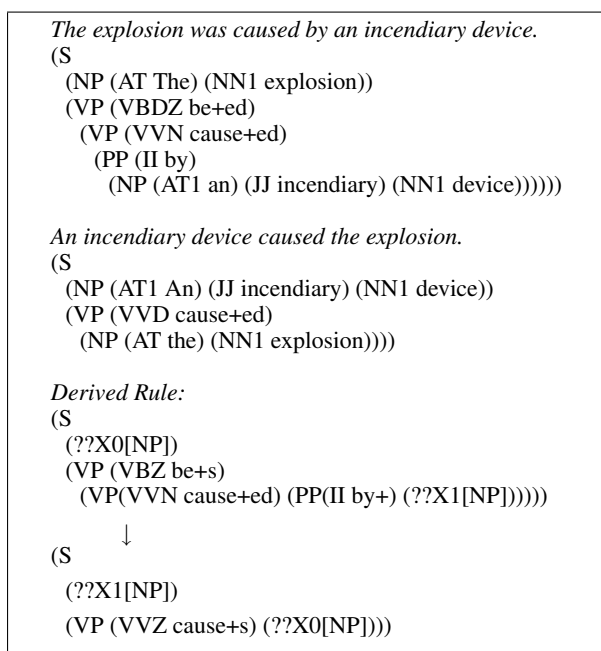
The corpus contains equal numbers of sentences from three different genres: PubMed Abstracts<sup>1</sup> (technical writing from the Biomedical domain), and articles from the British National Corpus<sup>2</sup> tagged as World News or Natural Science (popular science writing in the mainstream media).

### 3.2 Reformulation using Phrasal Parse Trees

As described above, we have access to a corpus that contains aligned sentences for each pair of types (a type is a combination of a discourse marker and an information order; thus we have 8 types). In principle it should be easy to learn transfer rules between parse trees of aligned sentences. Figure 1 shows parse trees (using the RASP parser (Briscoe et al., 2006)) for the active and the passive voice with “cause” as a verb. A transfer rule is derived by aligning nodes between two parse trees so that the rule only contains the differences in structure between the trees. In the representation in Figure 1, the variable ??X0[NP] maps

<sup>1</sup>PubMed URL: <http://www.ncbi.nlm.nih.gov/pubmed/>

<sup>2</sup>The British National Corpus, version 3 (BNC XML Edition). 2007. <http://www.natcorp.ox.ac.uk>



**Figure 1:** Example of a transfer rule derived from two parse trees.

onto any node (subtree) with label NP. RASP performs a morphological analysis of words (shown as lemma+suffix in the figure). Thus such rules can be used to account for changes in morphology, as in example 3a.–b. above.

In practise however, the parse tree representation is too dependent on the grammar rules employed by the parser. For instance, the parse tree for the sentence:

```

The explosion was presumed to be caused by an
incendiary device.
(S
  (NP (AT The) (NN1 explosion))
  (VP (VBDZ be+ed)
    (VP (VVN presume+ed)
      (VP (TO to)
        (VP (VB0 be)
          (VP (VVN cause+ed)
            (PP (II by) (NP (AT1 an)
              (JJ incendiary) (NN1 device))))))))))

```

looks very different and does not match the rule in figure 1. With longer sentences, further problems arise when similar strings are parsed differently in the two aligned sentences (for example, different PP attachment) – these lead to very complicated rules, often with more than 20 variables. We split our data into development/training (96 instances of passive to active) and test sets (48 instances of passive). Using the top parse for each sentence, we derived 92 rules, including the one shown in Figure 1. However, coverage of these rules over the test corpus was poor (less than 10% recall). By learning rules using the top 20 parses for each sentence rather than just the top parse,

we could improve coverage to around 70%, but this involved the acquisition of over 4000 different rules – just to change voice. The situation was even worse for reformulations that change syntactic categories, such as “because” to “cause”, and we obtained more than 20,000 rules that still gave us a coverage of only around 15% for the test set.

We concluded that this was not a sensible representation for general text reformulation. In other words, while substitution grammars for parse trees have been shown to be useful for sentence compression tasks (e.g., Cohn and Lapata (2009)), they are less useful for more complex lexico-syntactic reformulation tasks.

### 3.3 Reformulation using MRS

Another option is to use a bi-directional grammar and perform the transforms at a semantic level. We now briefly discuss the use of Minimal Recursion Semantics (MRS) as a representation for transfer rules. Consider a very short example for ease of illustration:

*Tom ate because of his hunger.*

This can be analysed by a deep grammar to give a compositional semantic representation which captures the information that is available from the syntax and inflectional morphology. We show this sentence below in the Minimal Recursion Semantics (MRS) (Copestake et al., 2005) representation, as produced by the English Resource Grammar (ERG<sup>3</sup>) (Flickinger, 2000), but considerably simplified for ease of exposition and to save space:

```

named(x5, Tom), _eat_v_1(e2, x5),
_because_of(e2, x11), poss(x11, x16),
pron(x16), _hunger_n(x11)

```

The main part of the MRS structure is a list of elementary predications (EPs), which may have predicates derived from lexemes (e.g., `_eat_v_1`; these are indicated by the leading underscore) or supplied by the grammar (e.g., `poss`). The ERG treats *because of* as a multiword expression and assigns it a semantics comparable to a preposition. Paraphrase rules map between semantic representations; for our application, a possible rule is the following:

```

_because_of(e, x), P(e, y) <->
_cause_v_1(e10, x, y, l1), l1:P(e, y)

```

Here ‘P’ is to be understood as a general predicate. The left hand side of the rule will match the preposition-like ‘because of’ relation when it has an event as an argument, where the event is the

<sup>3</sup>Available at <http://www.delph-in.net>.



characteristic event of an underspecified verbal EP. The right hand side indicates that the ‘because of’ can be substituted by a verbal relation corresponding to *cause*, with the verbal EP being a scopal argument. This rule matches the MRS above and maps it to the following (with  $P = \_eat\_v\_1$ ):

```
named(x5, Tom), l1:_eat_v_1(e2, x5),
_cause_v_1(e10, x11, x5, l1), poss(x11, x16),
pron(x16), _hunger_n(x11), x5 aeq x16
```

This can be input to the realiser, giving:

*His hunger caused Tom to eat.*

Writing transfer rules is intuitive and easy in MRS. Further, the use of a bi-directional grammar for generation ensures that the generated sentence is grammatical. An infrastructure of writing paraphrase rules exists in this framework and semantic transfer has also been explored for machine translation (e.g., Copestake et al. (1995)).

The problem we encountered, however, is that bidirectional grammars such as the ERG fail to parse ill-formed input and will also fail to analyse some well-formed input because of limitations in coverage of unusual constructions. Although the DELPH-IN parsing technology allows for unknown words, missing lexical items can also cause parse failure and even more problems for generation. The ERG gives an acceptable parse ‘out of the box’ for only around 50-60% of sentences from scientific papers. Further, the generator can get slow and memory intensive for long sentences and many of our sentences are around 30 words long. Much of this processing effort during generation is redundant as the input sentence can be used to narrow down generation choices, but as of now, the infrastructure does not exist to support this. Thus, while using a bi-directional grammar and semantic transfer might indeed be the most intuitive approach to complex lexico-syntactic reformulation, it is not quite feasible yet.

### 3.4 Reformulation using Typed Dependencies

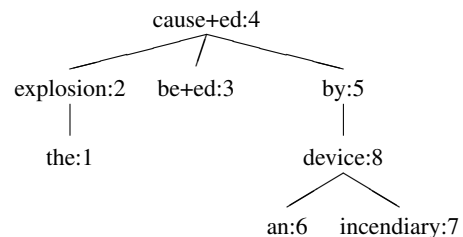
Having had mixed success with transforming phrasal parse trees and semantic representations, we turned our attention to typed dependency structures. We used the RASP toolkit (Briscoe et al., 2006) for finding grammatical relations (GRs) between words in the text. GRs are triplets consisting of a relation-type and arguments and also encode morphology (stem + suffix), word position (after colon) and part-of-speech (after underscore); GRs produced for the sentence:

*The explosion was caused by an incendiary device.*

are:

```
(\ncsubj| |cause+ed:4_VVN| |explosion:2_NN1| -)
(\aux| |cause+ed:4_VVN| |be+ed:3_VBDZ|)
(\passive| |cause+ed:4_VVN|)
(\iobj| |cause+ed:4_VVN| |by:5_II|)
(\dobj| |by:5_II| |device:8_NN1|)
(\det| |device:8_NN1| |an:6_AT1|)
(\ncmod| | |device:8_NN1| |incendiary:7_JJ|)
(\det| |explosion:2_NN1| |the:1_AT|)
```

This representation shares aspects of phrasal parse trees and MRS. Note that the sets of dependencies (such as those above) represent a tree.<sup>4</sup> While phrase structure trees such as those in Section 3.2 represent the nesting of constituents with the actual words at the leaf nodes, dependency trees have words at every node:



To generate from a dependency tree, we need to know the order in which to process nodes - in general tree traversal will be ‘inorder’; i.e. left subtrees will be processed before the root and right subtrees after. These are generation decisions that would usually be guided by the type of dependency and statistical preferences for word and phrase order. However, we can simply use the word positions (1–8) from the original sentence.

While typed dependencies share characteristics with parse trees, the flat structure represents dependencies between words, and we can write transformation rules for this representation in fairly compact form. For instance, a transformation rule to convert the above to active voice would require five deletions and two insertions:

1. Match and Delete:

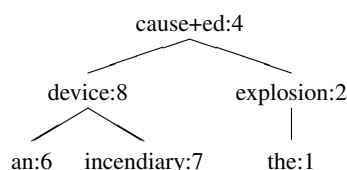
- (\passive| |??X0|)
- (\iobj| |??X0| |??X1(by\_II)|)
- (\dobj| |??X1| |??X2|)
- (\ncsubj| |??X0| |??X3| -)
- (\aux| |??X0| |??X4|)

2. Insert:

- (\ncsubj| |??X0| |??X2| -)
- (\dobj| |??X0| |??X3|)

<sup>4</sup>In fact, the GR scheme is only ‘almost’ acyclic. There are a small number of (predictable) relations that introduce cycles; for instance, dependencies between the head of a relative clause and the verb in the relative clause are represented as both a clausal modifier relation (cmod head verb) and an object relation (obj verb head). To resolve this, we use a fixed set of rules to remove these cycles from the dependency graph and ensure a tree structure.

Thus far, the rule looks very similar to rules written for MRS: one list of predicates is replaced by another. Applying this transformation to the GR set above creates a new dependency tree:



However, unlike the case with MRS, where a statistical generator decides issues of morphology and ordering, we have to specify the consequences of the rule application for generation. Note that we can no longer rely on the original word order to determine the order in which to traverse the tree for generation. Thus our transformation rules, in addition to Deletion and Insertion operations, also need to provide rules for tree traversal order. These only need to be provided for nodes where the transform has reordered subtrees (“??X0”, which instantiates to “cause+ed:4” in the trees). Our rule would thus include:

### 3. Traversal Order Specifications:

- (a) Node ??X0: [??X2, ??X0, ??X3]

This states that for node ??X0, the traversal order should be subtree ??X2 followed by current node ??X0 followed by subtree ??X3. Using this specification would allow us to traverse the tree using the original word order for nodes with no order specification, and the specified order where a specification exist. In the above instance, this would lead us to generate:

*An incendiary device caused the explosion.*

Our transfer rule is still incomplete and there is one further issue that needs to be addressed – operations to be performed on nodes rather than relations. There are two node-level operations that might be required for sentence reformulation:

**1. Lexical substitution:** In our example above, we still need to ensure number agreement for the verb “cause” (??X0). By changing voice, ??X0 now has to agree with ??X2 rather than ??X3. Further the tense of ??X0 was encoded in the auxiliary verb ??X4 that has been deleted from the GRs. We thus need the transfer rule to encode the lexical substitution required for node ??X0:

### 4. Lexical substitution:

- (a) Node ??X0: IF (??X4 is Present Tense) THEN { IF (??X2 is Plural) THEN {SET ??X0:SUFFIX =“s”} ELSE {SET ??X0:SUFFIX =“s”} }

Other lexical substitutions are easier to specify; for instance to reformulate “*John ran because David shouted.*” as “*David’s shouting caused John to run*”, the following lexical substitution rule is required for node ??Xn representing “shout” that replaces its suffix “ed” with “ing”:

Lexical substitution: Node ??Xn: Suffix=“ing”

**2. Node deletion:** This is an operation that removes a node from the tree. Any subtrees are moved to the parent node. If a root node is deleted, one of the children adopts the rest. By default, the right-most child takes the rest as dependents, but we allow the rule to specify the new parent. In the above example, we want to remove the nodes ??X1 (“by”) and ??X4 (“was”) (note that deleting a relation does not necessarily remove a node – there might be other nodes connected to ??X1 or ??X4). We would like to move these to the node ??X0 (“cause”):

### 5. Node Deletion:

- (a) Node ??X1: Target=??X0  
(b) Node ??X4: Target=??X0

Node deletion is easily implemented using search and replace on sets of GRs. It is central to reformulations that alter syntactic categories of discourse markers; for instance, to reformulate “*The cause of X is Y*” as “*Y causes X*”, we need to delete the verb “is” and move its dependents to the new verb “causes”.

To summarise, we propose a framework for lexico-syntactic reformulation based on typed dependency structures and have discussed the form of a transformation. We now specify the structure of transfer rules and tree nodes more formally.

## Specification for Transfer Rules

Our proposal is based on applying transfer rules to lists of grammatical relations (GRs). Our transfer rules take the form of five lists:

1. CONTEXT: Transform only proceeds if this list of GRs can be unified with the input GRs.
2. DELETE: List of GRs to delete from input.
3. INSERT: List of GRs to insert into input.
4. ORDERING: List of nodes with subtree order specified
5. NODE-OPERATIONS: List of lexical substitutions and deletion operations on nodes.

For the reformulations in this paper, the CONTEXT and DELETE lists are one and the same, but one can imagine reformulation tasks where extra context needs to be specified to determine whether reformulation is appropriate. The first three lists

correspond to the CONTEXT, INPUT and OUTPUT lists used to specify transform in the MRS framework. However, because we do not use a formal grammar for generation, we need two further lists that capture changes in morphology or constituent ordering. The list ORDERING is used to traverse the dependency tree constructed from the transformed GRs. Again, the lexical substitution lists are prescriptions for generation. We restrict our lexical substitutions to change of suffix and part of speech (for instance, “X is a *frequent* cause of Y” to “X *frequently* causes Y”), but in general this can be an arbitrary string substitution (for instance, “X and Y are *two* causes of Z” to “X and Y *both* cause Z”).

In this paper, we have tried to do away with a generator altogether by encoding generation decisions within the transfer rule. A case can be made, particularly for the issue of agreement, for such issues to be handled by a generator. This would make the transfer rules simpler to write, and easier to learn automatically in a supervised setting.

### Specification for Dependency Tree

Applying a transfer rule specified above results in a new set of GRs. To generate a sentence, we need to create a dependency tree from these GRs. As described earlier, a dependency tree needs to be traversed “inorder” to generate a sentence. This means that at each node, the order in which to visit the daughters and the current node needs to be specified. To enable this, we propose that each node in the tree have the following features:

1. VALUE: stem, suffix and part-of-speech of the word;
2. PARENT: parent node;
3. CHILDREN: list of daughters;
4. ORDER: list specifying order in which to visit children and current node.

The parent node is required for DELETE operations and to find the root of the tree (node with no parent). Further, if there is more than one node with no parent, the GRs do not form a tree and generation will result in multiple fragments.

The dependency tree is constructed using the following algorithm:

1. For each word in the list of GRs:
  - (a) Create a Node and instantiate the VALUE field.
2. For each GR (relation word1 word2):
  - (a) If GR is one that introduces a cycle, remove it from list, else add the node created for word2 to the CHILDREN list of node for word1 and set PARENT of word 2 to word1.
3. After Step 2, the tree is created. Now for each Node:

- (a) If an ORDERING specification is introduced for this node by the transformation rule, copy that list to the ORDER field, else add the daughter nodes to the ORDER list in increasing order of word position.

The reformulated sentence is generated by traversing the tree “inorder”, outputting the word at each node visited (the stem, suffix and part-of-speech tag are fed to the RASP morphological generator, which returns the correct word).

## 4 Evaluating Transformation Rules

In this paper we have proposed a framework for complex lexico-syntactic reformulations. We want to evaluate our framework for (a) how easy it is to write transformation rules, (b) how many are required for intuitive lexico-syntactic reformulations and (c) how robust the transformation is to parsing errors. With this intended purpose, we evaluate hand-written transformation rules that have been developed looking at one third of the corpus (48 sentences) and tested on the remaining two thirds (96 sentences). We report results using:

- **Recall:** The proportion of sentences in the test set for which a transform was performed; i.e., (a) the DELETE pattern matched the input GRs and (b) there was exactly one root node in the transformed GRs resulting in exactly one sentence being output
- **Precision:** The proportion of transformed sentence that were accurate; i.e., grammatical with (a) correct verb agreement and inflexion and (b) modifiers/complements appearing in acceptable orders.

Note that we are merely evaluating the framework and not evaluating the utility of these transformations for text simplification – that would require an evaluation using test subjects drawn from our intended users. Table 1 provides some examples of accurate and inaccurate transformations.

The rule for converting passives to actives described in Section 3.4 already achieves a recall of 42% and precision of 83%. Writing 6 additional rules to handle reduced relative clauses (1a-b, Table 1) etc., we could boost recall to 71% with precision dropping marginally to 82%. We hand-crafted rules to implement three other reformulations. These were selected based on results from the Siddharthan and Katsos (2010) study that suggested:

1. cause as a noun (either information ordering), passive voice, “because of” and “because a, b” formulations (versions b,d,e,f,g and h in Example 1, Section 1) are dispreferred by lay readers. Moreover, these are common constructs in scientific writing.
2. cause as a verb in active voice and “b because a” are the most preferred formulations for lay readers.

Accurate Transformations	
1a.	Apart from occasional problems of ensemble caused by the complex rhythms of the outer movements, the orchestra gave an animated and committed reading of the work. [B-CAUSEBY-A→A-CAUSE-B]
b.	Apart from occasional problems of ensemble the complex rhythms of the outer movements caused, the orchestra gave an animated and committed reading of the work.
2a.	Because of transvection, the expression of a gene can be sensitive to the proximity of a homolog. [BEC-OF-A-B→A-CAUSE-B]
b.	Transvection can cause the expression of a gene to be sensitive to the proximity of a homolog.
3a.	Because each myosin is expressed in Drosophila indirect flight muscle, in the absence of other myosin isoforms, this allows for muscle mechanical and whole organism locomotion assays. [BEC-A-B→B-BEC-A]
b.	In the absence of other myosin isoforms, this allows for muscle mechanical and whole organism locomotion assays because each myosin is expressed in Drosophila indirect flight muscle.
4a.	Almost certainly, however, the underlying cause of the war was the problem of Aquitaine. [CAUSEOF-B-A→A-CAUSE-B]
b.	Almost certainly, however, the underlying problem of Aquitaine caused the war.
Inaccurate Transformation	
5a.	Moreover, main road traffic has scarcely been slowed and concern should be caused by the rising number of cyclist casualties. [B-CAUSEBY-A→A-CAUSE-B]
b.	Moreover, the rising number of cyclist casualties should cause main road traffic has scarcely been slowed and concern.
6a.	Because of the risk of injury and the need to kill prey quickly, predators usually predate animals smaller than themselves. [BEC-OF-A-B→A-CAUSE-B]
b.	The risk of injury and the need cause kill to prey quickly predators usually predate animals smaller than themselves.

Table 1: Examples of automatic reformulations (version a. is the original and b. the reformulation).

Handcrafted rules	n	P	R	F
B-CAUSEBY-A → A-CAUSE-B	7	.82 (1.00)	.71 (.75)	.76 (.86)
BEC-OF-A-B → A-CAUSE-B	9	.75 (.92)	.70 (1.00)	.72 (.97)
BEC-A-B → B-BEC-A	8	.85 (.92)	.83 (.87)	.84 (.89)
CAUSEOF → A-CAUSE-B	6	.97 (.90)	.78 (1.00)	.86 (.95)

Table 2: Number of Rules (n), Precision, Recall and F-Measure for lexico-syntactic reformulation using hand-crafted rules over GRs. Numbers in brackets are over the subset of the corpus that contains only the original sentences from PubMed and the BNC.

We summarise our results in Table 2. Most of the sentences in the corpus are manual reformulations and some of them are quite stilted. The numbers in brackets show performance over the smaller set of original sentences from PubMed and the BNC. These are more indicative of how the rules will perform on real data. Our results suggest that the framework we propose is adequate for a range of lexico-syntactic reformulations and a fairly small number of rules is required to capture a reformulation.

Loss of recall was usually from parsing error (either misparses, in which case our rules don't match the GRs, or partial parses, where a full tree can't be formed because of missing GRs).

Loss of precision was a more worrying issue as it often resulted in badly corrupted output. This was usually the result of either bad parser decisions regarding attachment or scope or just misparsing (e.g., wide scoping of “and” in 5a-b and parsing “prey” as a verb in 6a-b, Table 1). It might be possible to trade-off recall for improved precision by identifying sentences where ambiguity is a problem (by looking at multiple parses).

## 5 Conclusions and Future Work

In this paper we have reported our experience with using different linguistic formalisms as representations for applying transform rules to generate complex lexico-syntactic reformulations of sentences expressing the discourse relation of causation. We find typed dependency structures to be the most suited for this task and report that hand-crafted transformation rules generalise well to sentences in an unseen test corpus. We believe that the framework we have described is adequate for a range of regeneration tasks focused on text simplification. While in this paper we focus on the discourse relation of causation, other discourse relations commonly used in scientific writing can also be realised using markers with different lexico-syntactic properties; for instance, *contrast* can be expressed using markers such as “while”, “unlike”, “but”, “compared to”, “in contrast to” and “the difference between”. Our rules for voice conversion and information reordering for subordination are already general enough to be applied to non-causal constructs. We also plan to use our framework to explore sentence simplification and sentence shortening applications.

We would in the future like to learn transformations rules automatically from a corpus. Hand-crafting can get tedious as there are 17 types of grammatical relations to take into account in the RASP scheme. Preliminary work by us in this regard suggests that augmenting a few hand-crafted rules with around a hundred automatically learnt rules can increase recall substantially. However, our learning framework as yet does not allow node transformations, and more work is required here.

## Acknowledgements

This work was supported by the Economic and Social Research Council (Grant Number RES-000-22-3272). We would also like to thank Dan Flickinger and Ann Copestake for many discussions on the topic of paraphrase and for help with using the ERG.

## References

- R.C. Anderson and A. Davison. 1988. Conceptual and empirical bases of readability formulas. In Alice Davison and G. M. Green, editors, *Linguistic Complexity and Text Comprehension: Readability Issues Reconsidered*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23.
- I.L. Beck, M.G. McKeown, G.M. Sinatra, and J.A. Loxterman. 1991. Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility. *Reading Research Quarterly*, pages 251–276.
- T. Briscoe, J. Carroll, and R. Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL*, volume 6.
- J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10, Madison, Wisconsin.
- T. Cohn and M. Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34(1):637–674.
- A. Copestake, D. Flickinger, R. Malouf, S. Riehemann, and I. Sag. 1995. Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 15–32.
- A. Copestake, D. Flickinger, I. Sag, and C. Pollard. 2005. Minimal recursion semantics: An introduction. *Research in Language and Computation*, 3:281–332.
- D. Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.
- M. Galley and K. McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *HLT-NAACL 2007: Main Proceedings*, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.
- A. Ibrahim, B. Katz, and J. Lin. 2003. Extracting paraphrases from aligned corpora. In *Proceedings of The Second International Workshop on Paraphrasing*.
- N. Kaji, D. Kawahara, S. Kurohashi, and S. Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 215–222, Philadelphia, USA.
- K. Knight and D. Marcu. 2000. Statistics-based summarization — step one: Sentence compression. In *Proceeding of The American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 703–710.
- J.J. L'Allier. 1980. *An evaluation study of a computer-based lesson that adjusts reading level by monitoring on task reader characteristics*. Ph.D. thesis, University of Minnesota, Minneapolis, MN.
- E.T. Levy. 2003. The roots of coherence in discourse. *Human Development*, pages 169–88.
- T. Linderholm, M.G. Everson, P. van den Broek, M. Mischinski, A. Crittenden, and J. Samuels. 2000. Effects of Causal Text Revisions on More-and Less-Skilled Readers' Comprehension of Easy and Difficult Texts. *Cognition and Instruction*, 18(4):525–556.
- L. G. M. Noordman and W. Vonk. 1992. Reader's knowledge and the control of inferences in reading. *Language and Cognitive Processes*, 7:373–391.
- R. Power, D. Scott, and N. Bouayad-Agha. 2003. Generating texts with style. *Proceedings of the 4th International Conference on Intelligent Texts Processing and Computational Linguistics*.
- S. Riezler, T.H. King, R. Crouch, and A. Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *HLT-NAACL 2003: Main Proceedings*, Edmonton, Canada.
- A. Siddharthan and N. Katsos. 2010. Reformulating discourse connectives for non-expert readers. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, Los Angeles, CA.
- A. Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.
- S. Williams and E. Reiter. 2008. Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, 14(04):495–525.
- S. Williams, E. Reiter, and L. Osman. 2003. Experiments with discourse-level choices and readability. In *Proceedings of the European Natural Language Generation Workshop (ENLG), EACL'03*, pages 127–134, Budapest, Hungary.



# Towards an Extrinsic Evaluation of Referring Expressions in Situated Dialogs

Philipp SPANGER IIDA Ryu TOKUNAGA Takenobu

{philipp,ryu-i,take}@cl.cs.titech.ac.jp

TERAI Asuka KURIYAMA Naoko

asuka@nm.hum.titech.ac.jp kuriyama@hum.titech.ac.jp

Tokyo Institute of Technology

## Abstract

In the field of referring expression generation, while in the static domain both intrinsic and extrinsic evaluations have been considered, extrinsic evaluation in the dynamic domain, such as in a situated collaborative dialog, has not been discussed in depth. In a dynamic domain, a crucial problem is that referring expressions do not make sense without an appropriate preceding dialog context. It is unrealistic for an evaluation to simply show a human evaluator the whole period from the beginning of a dialog up to the time point at which a referring expression is used. Hence, to make evaluation feasible it is indispensable to determine an appropriate shorter context. In order to investigate the context necessary to understand a referring expression in a situated collaborative dialog, we carried out an experiment with 33 evaluators and a Japanese referring expression corpus. The results contribute to finding the proper contexts for extrinsic evaluation in dynamic domains.

## 1 Introduction

In recent years, the NLG community has paid significant attention to the task of generating referring expressions, reflected in the setting-up of several competitive events such as the TUNA and GIVE-Challenges at ENLG 2009 (Gatt et al., 2009; Byron et al., 2009).

With the development of increasingly complex generation systems, there has been heightened interest in and an ongoing significant discussion on different evaluation measures for referring expressions. This discussion is carried out broadly in the field of generation, including in the multi-modal domain, e.g. (Stent et al., 2005; Foster, 2008).

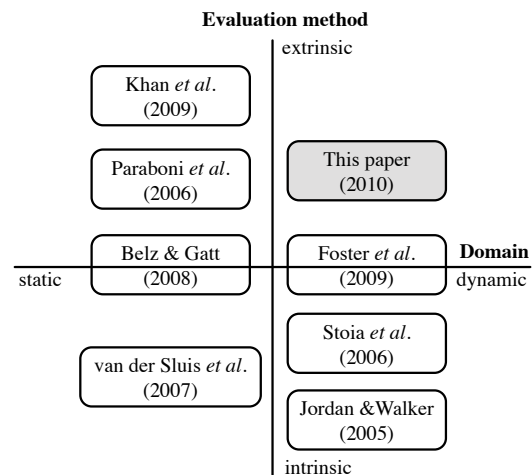


Figure 1: Overview of recent work on evaluation of referring expressions

Figure 1 shows a schematic overview of recent work on evaluation of referring expressions along the two axes of evaluation method and domain in which referring expressions are used.

There are two different evaluation methods corresponding to the bottom and the top of the vertical axis in Figure 1: *intrinsic* and *extrinsic* evaluations (Sparck Jones and Galliers, 1996). Intrinsic methods often measure similarity between the system output and the gold standard corpora using metrics such as tree similarity, string-edit-distance and BLEU (Papineni et al., 2002). Intrinsic methods have recently become popular in the NLG community. In contrast, extrinsic methods evaluate generated expressions based on an external metric, such as its impact on human task performance.

While intrinsic evaluations have been widely used in NLG, e.g. (Reiter et al., 2005), (Cahill and van Genabith, 2006) and the competitive 2009 TUNA-Challenge, there have been a number of criticisms against this type of evaluation. (Reiter

and Sripada, 2002) argue, for example, that generated text might be very different from a corpus but still achieve the specific communicative goal.

An additional problem is that corpus-similarity metrics measure how well a system reproduces what speakers (or writers) do, while for most NLG systems ultimately the most important consideration is its effect on the human user (i.e. listener or reader). Thus (Khan et al., 2009) argues that “measuring *human-likeness* disregards effectiveness of these expressions”.

Furthermore, as (Belz and Gatt, 2008) state “there are no significant correlations between intrinsic and extrinsic evaluation measures”, concluding that “similarity to human-produced reference texts is not necessarily indicative of quality as measured by human task performance”.

From early on in the NLG community, task-based extrinsic evaluations have been considered as the most meaningful evaluation, especially when having to convince people in other communities of the usefulness of a system (Reiter and Belz, 2009). Task performance evaluation is recognized as the “only known way to measure the effectiveness of NLG systems with real users” (Reiter et al., 2003). Following this direction, the GIVE-Challenges (Koller et al., 2009) at INLG 2010 (instruction generation) also include a task-performance evaluation.

In contrast to the vertical axis of Figure 1, there is the horizontal axis of the domain in which referring expressions are used. Referring expressions can thus be distinguished according to whether they are used in a *static* or a *dynamic* domain, corresponding to the left and right of the horizontal axis of Figure 1. A static domain is one such as the TUNA corpus (van Deemter, 2007), which collects referring expressions based on a motionless image. In contrast, a dynamic domain comprises a constantly changing situation where humans need context information to identify the referent of a referring expression.

Referring expressions in the static domain have been evaluated relatively extensively. A recent example of an intrinsic evaluation is (van der Sluis et al., 2007), who employed the Dice-coefficient measuring corpus-similarity. There have been a number of extrinsic evaluations as well, such as (Paraboni et al., 2006) and (Khan et al., 2009), respectively measuring the effect of overspecification on task performance and the impact of gener-

ated text on accuracy as well as processing speed. They belong thus in the top-left quadrant of Figure 1.

Over a recent period, research in the generation of referring expressions has moved to dynamic domains such as situated dialog, e.g. (Jordan and Walker, 2005) and (Stoia et al., 2006). However, both of them carried out an intrinsic evaluation measuring corpus-similarity or asking evaluators to compare system output to expressions used by human (the right bottom quadrant in Figure 1).

The construction of effective generation systems in the dynamic domain requires the implementation of an extrinsic task performance evaluation. There has been work on extrinsic evaluation of instructions in the dynamic domain on the GIVE-2 challenge (Byron et al., 2009), which is a task to generate instructions in a virtual world. It is based on the GIVE-corpus (Gargett et al., 2010), which is collected through keyboard interaction. The evaluation measures used are e.g. the number of successfully completed trials, completion time as well as the numbers of instructions the system sent to the user. As part of the JAST project, a Joint Construction Task (JCT) puzzle construction corpus (Foster et al., 2008) was created which is similar in some ways in its set-up to the REX-J corpus which we use in the current research. There has been some work on evaluating generation strategies of instructions for a collaborative construction task on this corpus, both considering intrinsic as well as extrinsic measures (Foster et al., 2009). Their main concern is, however, the interaction between the text structure and usage of referring expressions. Therefore, their “context” was given a priori.

However, as can be seen from Figure 1, in the field of referring expression generation, while in the static domain both intrinsic and extrinsic evaluations have been considered, the question of realizing an extrinsic evaluation in the dynamic domain has not been dealt with in depth by previous work. This paper addresses this shortcoming of previous work and contributes to “filling in” the missing quadrant of Figure 1 (the top-right).

The realization of such an extrinsic evaluation faces one key difficulty. In a static domain, an extrinsic evaluation can be realized relatively easily by showing evaluators the *static* context (e.g. any image) and a referring expression, even though this is still costly in comparison to intrinsic meth-



ods (Belz and Gatt, 2008).

In contrast, an extrinsic evaluation in the *dynamic* domain needs to present an evaluator with the *dynamic* context (e.g. a certain length of the recorded dialog) preceding a referring expression. It is clearly not feasible to simply show the *whole* preceding dialog; this would make even a very small-scale evaluation much too costly. Thus, in order to realize a cost-effective extrinsic evaluation in a dynamic domain we have to deal with the additional parameter of time length and content of the context shown to evaluators.

This paper investigates the context necessary for humans to understand different types of referring expressions in a situated domain. This work thus charts new territory and contributes to developing an extrinsic evaluation in a dynamic domain. Significantly, we consider not only linguistic but also extra-linguistic information as part of the context, such as the actions that have been carried out in the preceding interaction. Our results indicate that, at least in this domain, extrinsic evaluation results in dynamic domains can depend on the specific amount of context shown to the evaluator. Based on the results from our evaluation experiments, we discuss the broader conclusions to be drawn and directions for future work.

## 2 Referring Expressions in the REX-J Corpus

We utilize the REX-J corpus, a Japanese corpus of referring expressions in a situated collaborative task (Spanger et al., 2009a). It was collected by recording the interaction of a pair of dialog participants solving the Tangram puzzle cooperatively. The goal of the Tangram puzzle is to construct a given shape by arranging seven pieces of simple figures as shown in Figure 2

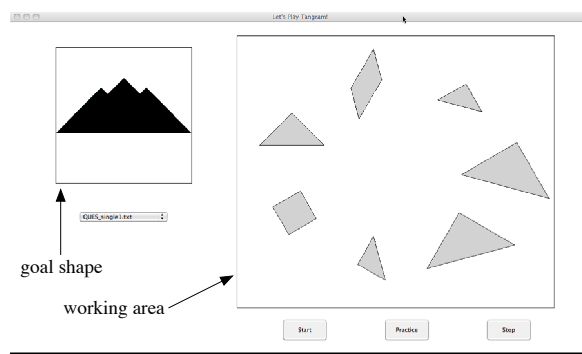


Figure 2: Screenshot of the Tangram simulator

In order to record the precise position of every piece and every action by the participants, we implemented a simulator. The simulator displays two areas: a goal shape area, and a working area where pieces are shown and can be manipulated.

We assigned different roles to the two participants of a pair: *solver* and *operator*. The solver can see the goal shape but cannot manipulate the pieces and hence gives instructions to the operator; by contrast, the operator can manipulate the pieces but can not see the goal shape. The two participants collaboratively solve the puzzle sharing the working area in Figure 2.

In contrast to other recent corpora of referring expressions in situated collaborative tasks (e.g. COCONUT corpus (Di Eugenio et al., 2000) and SCARE corpora (Byron et al., 2005)), in the REX-J corpus we allowed comparatively large real-world flexibility in the actions necessary to achieve the task (such as flipping, turning and moving of puzzle pieces at different degrees), relative to the task complexity. The REX-J corpus thus allows us to investigate the interaction of linguistic and extra-linguistic information. Interestingly, the GIVE-2 challenge at INLG 2010 notes its “main novelty” is allowing “continuous moves rather than discrete steps as in GIVE-1”. Our work is in line with the broader research trend in the NLG community of trying to get away from simple “discrete” worlds to more realistic settings.

The REX-J corpus contains a total of 1,444 tokens of referring expressions in 24 dialogs with a total time of about 4 hours and 17 minutes. The average length of each dialog is 10 minutes 43 seconds. The asymmetric data-collection setting encouraged referring expressions from the solver (solver: 1,244 tokens, operator: 200 tokens). We exclude from consideration 203 expressions referring to either groups of pieces or whose referent cannot be determined due to ambiguity, thus leaving us 1,241 expressions.

We identified syntactic/semantic features in the collected referring expressions as listed in Table 1: (a) demonstratives (adjectives and pronouns), (b) object attribute-values, (c) spatial relations and (d) actions on an object. The underlined part of the examples denotes the feature in question.

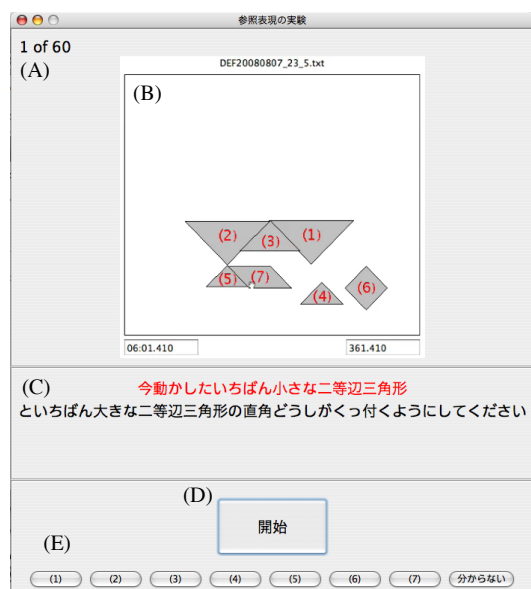
## 3 Design of Evaluation Experiment

The aim of our experiment is to investigate the “context” (content of the time span of the recorded

Table 1: Syntactic and semantic features of referring expressions in the REX-J corpus

Feature	Tokens	Example
(a) demonstrative	742	<i>ano migigawa no sankakkei</i> ( <u>that</u> triangle at the right side)
(b) attribute	795	<i>tittyai sankakkei</i> (the <u>small</u> triangle)
(c) spatial relations	147	<i>hidari no okkii sankakkei</i> (the small triangle <u>on the left</u> )
(d) action-mentioning	85	<i>migi ue ni doketa sankakkei</i> (the triangle you <u>put away</u> to the top right)

interaction prior to the uttering of the referring expression) necessary to enable successful identification of the referent of a referring expression. Our method is to vary the context presented to evaluators and then to study the impact on human referent identification. In order to realize this, for each instance of a referring expression, we vary the length of the video shown to the evaluator.



(A): Counter (1-60)  
 (B): Video of shared working area in the simulator  
 (C): Utterance including the referring expression to evaluate (shown in red)  
 (D): Start/repeat button  
 (E): Selection buttons (1-7) and "I don't know"-button

Figure 3: The interface presented to evaluators

The basic procedure of our evaluation experiment is as follows:

- (1) present human evaluators with speech and video from a dialog that captures shared working area of a certain length previous to

- the uttering of a referring expression,
- (2) stop the video and display as text the next solver's utterance including the referring expression (shown in red),
- (3) ask the evaluator to identify the referent of the presented referring expression (if the evaluator wishes, he/she can replay the video as many times as he likes),
- (4) proceed to the next referring expression (go to (1)).

Figure 3 shows a screenshot of the interface prepared for this experiment.

The test data consists of three types of referring expressions: DPs (demonstrative pronouns), AMEs (action-mentioning expressions), and OTHERs (any other expression that is neither a DP nor AME, e.g. intrinsic attributes and spatial relations). DPs are the most frequent type of referring expression in the corpus. AMEs are expressions that utilize an action on the referent such as "the triangle you put away to the top right" (see Table 1)<sup>1</sup>. As we pointed out in our previous paper (Spanger et al., 2009a), they are also a fundamental type of referring expression in this domain.

The basic question in investigating a suitable context is what information to consider about the preceding interaction; i.e. over what parameters to vary the context. In previous work on the generation of demonstrative pronouns in a situated domain (Spanger et al., 2009b), we investigated the role of linguistic and extra-linguistic information, and found that time distance from the last action (LA) on the referent as well as the last mention (LM) to the referent had a significant influence on the usage of referring expressions. Based on those results, we focus on the information on the referent, namely LA and LM.

For both AMEs and OTHERs, we only consider two possibilities of the order in which LM and LA appear before a referring expression (REX), depending on which comes first. These are shown in Figure 4, context patterns (a) LA-LM and (b) LM-LA. Towards the very beginning of a dialog, some referring expressions have no LM and LA; those expressions are not considered in this research.

All instances of AMEs and OTHERs in our test data belong to either the LA-LM or the LM-LA

<sup>1</sup>An action on the referent is usually described by a verb as in this example. However, there are cases with a verb ellipsis. While this would be difficult in English, it is natural and grammatical in Japanese.

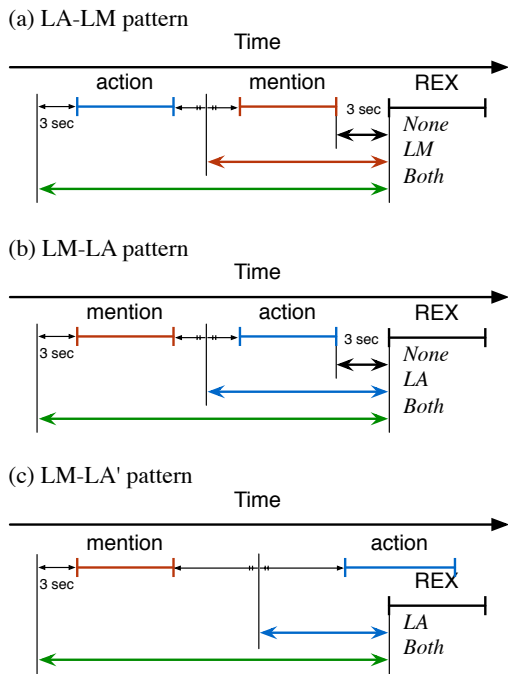


Figure 4: Schematic overview of the three context Patterns

pattern. For each of these two context patterns, there are three possible contexts<sup>2</sup>: *Both* (including both LA and LM), *LA/LM* (including either LA or LM) and *None* (including neither). Depending on the order of LA and LM prior to an expression, only one of the variations of *LA/LM* is possible (see Figure 4 (a) and (b)).

In contrast, DPs tend to be utilized in a deictic way in such situated dialogs (Piwek, 2007). We further noted in (Spanger et al., 2009b), that DPs in a collaborative task are also frequently used when the referent is under operation. While they belong neither to the LA-LM nor the LM-LA pattern, it would be inappropriate to exclude those cases. Hence, for DPs we consider another situation where the last action on the referent *overlaps* with the utterance of the DP (Figure 4 (c) LM-LA' pattern). In this case, we consider an ongoing operation on the referent as a “last action”. Another peculiarity of the LM-LA' pattern is that we have no *None* context in this case, since there is no way to show a video without showing LA (the current operation).

Given the three basic variations of context, we recruited 33 university students as evaluators and

<sup>2</sup>To be more precise, we set a margin at the beginning of contexts as shown in Figure 4.

divided them equally into three groups, i.e. 11 evaluators per group. As for the referring expressions to evaluate, we selected 60 referring expressions used by the solver from the REX-J corpus (20 from each category), ensuring all were correctly understood by the operator during the recorded dialog. We selected those 60 instances from expressions where both LM and LA appeared within the last 30 secs previous to the referring expression. This selection excludes initial mentions, as well as expressions where only LA or only LM exists or they do not appear within 30 secs. Hence the data utilized for this experiment is limited in this sense. We need further experiments to investigate the relation between the time length of contexts and the accuracy of evaluators. We will return to this issue in the conclusion.

We combined 60 referring expressions and the three contexts to make the test instances. Following the Latin square design, we divided these test instances into three groups, distributing each of the three contexts for every referring expression to each group. The number of contexts was uniformly distributed over the groups. Each instance group was assigned to each evaluator group.

For each referring expression instance, we record whether the evaluator was able to correctly identify the referent, how long it took them to identify it and whether they repeated the video (and if so how many times).

Reflecting the distribution of the data available in our corpus, the number of instances per context pattern differs for each type of referring expression. For AMEs, overwhelmingly the last action on the referent was more recent than the last mention. Hence we have only two LA-LM patterns among the 20 AMEs in our data. For OTHERs, the balance is 8 to 12, with a slight majority of LM-LA patterns. For DPs, there is a strong tendency to use a DP when a piece is under operation (Spanger et al., 2009b). Of the 20 DPs in the data, 2 were LA-LM, 5 were LM-LA pattern while 13 were of the LM-LA' pattern (i.e. their referents were under operation at the time of the utterance). For these 13 instances of LM-LA' we do not have a *None* context.

The average stimulus times, i.e. time period of presented context, were 7.48 secs for *None*, 11.04 secs for *LM/LA* and 18.10 secs for *Both*.

Table 2: Accuracy of referring expression identification per type and context

Type	context pattern \ Context	<i>None</i>	<i>LM/LA</i>	<i>Both</i>	Increase [ <i>None</i> → <i>Both</i> ]
DP	(LA-LM)	0.909 (20/22)	0.955 (21/22)	0.955 (21/22)	0.046
	(LM-LA)	0.455 (25/55)	0.783 (155/198)	0.843 (167/198)	0.388
Total		0.584	0.800	0.855	0.271
AME	(LA-LM)	0.227 (5/22)	0.455 (10/22)	0.682 (15/22)	0.455
	(LM-LA)	0.530 (105/198)	0.859 (170/198)	0.879 (174/198)	0.349
Total		0.500	0.818	0.859	0.359
OTHER	(LA-LM)	0.784 (69/88)	0.852 (75/88)	0.943 (83/88)	0.159
	(LM-LA)	0.765 (101/132)	0.788 (104/132)	0.879 (116/132)	0.114
Total		0.773	0.814	0.905	0.132
Overall		0.629 (325/517)	0.811 (535/660)	0.903 (576/638)	0.274

## 4 Results and Analysis

In this section we discuss the results of our evaluation experiment. In total 33 evaluators participated in our experiment, each solving 60 problems of referent identification. Taking into account the absence of the *None* context for the DPs of the LM-LA’ pattern (see (c) in Figure 4), we have 1,815 responses to analyze. We focus on the impact of the three contexts on the three types of referring expressions, considering the two context patterns LA-MA and LM-LA.

### 4.1 Overview of Results

Table 2 shows the micro averages of the accuracies of referent identification of all evaluators over different types of referring expressions with different contexts. Accuracies increase with an increase in the amount of information in the context; from *None* to *Both* by between 13.2% (OTHERs) and 35.9% (AMEs). The average increase of accuracy is 27.4%.

Overall, for AMEs the impact of the context is the greatest, while for OTHERs it is the smallest. This is not surprising given that OTHERs tend to include intrinsic attributes of the piece and its spatial relations, which are independent of the preceding context. We conducted ANOVA with the context as the independent variable, testing its effect on identification accuracy. The main effect of the context was significant on accuracy with  $F(2, 1320) = 9.17$ ,  $p < 0.01$ . Given that for DPs we did not have an even distribution between contexts, we only utilized the results of AMEs and

OTHERs.

There are differences between expression types in terms of the impact of addition of LM/LA into the context, which underlines that when studying context, the relative role and contribution of LA and LM (and their interaction) must be looked at in detail for different types of referring expressions.

Over all referring expressions, the addition into a *None* context of LM yields an average increase in accuracy of 9.1% for all referring expression types, while for the same conditions the addition of LA yields an average increase of 21.3%. Hence, interestingly for our test data, the addition of LA to the context has a positive impact on accuracy by more than two times over the addition of LM.

It is also notable that even with neither LA nor LM present (i.e. the *None* context), the evaluators were still able to correctly identify referents in between 50–68.6% (average: 62.9%) of the cases. While this accuracy would be insufficient for the evaluation of machine generated referring expressions, it is still higher than one might expect and further investigation of this case is necessary.

### 4.2 Demonstrative Pronouns

For DPs, there is a very clear difference between the two patterns (LM-LA and LA-LM) in terms of the increase of accuracy with a change of context. While accuracy for the LA-LM pattern remains at a high level (over 90%) for all three contexts (and there is only a very small increase from *None* to *Both*), for the LM-LA pattern there is a strong increase from *None* to *Both* of 38.8%.

The difference in accuracy between the two

context patterns of DPs in the *None* context might come from the mouse cursor effect. The two expressions of LA-LM pattern happened to have a mouse cursor on the referent, when they were used, resulting in high accuracy. On the other hand, 4 out of 5 expressions of LM-LA pattern did not have a mouse cursor on the referent. We have currently no explanation for the relation between context patterns and the mouse position. While we have only 7 expressions in the *None* context for DPs and hence cannot draw any decisive conclusions, we note that the impact of the mouse position is a likely factor.

For the LM-LA pattern, there is an increase in accuracy of 32.8% from *None* to the *LA*-context. Overwhelmingly, this represents instances in which the referents are being operated at the point in time when the solver utters a DP (this is in fact the LM-LA' pattern, which has no *None* context). For those instances, the current operation information is sufficient to identify the referents. In contrast, addition of LM leads only to a small increase in accuracy of 5.6%. This result is in accordance with our previous work on the generation of DPs, which stressed the importance of extra-linguistic information in the framework of considering the interaction between linguistic and extra-linguistic information.

### 4.3 Action-mentioning Expressions

While for AMEs the number of instances is very uneven between patterns (similar to the distribution for DPs), there is a strong increase in accuracy from the *None* context to the *Both* context for both patterns (between 30% to almost 50%). However, there is a difference between the two patterns in terms of the relative contribution of LM and LA to this increase.

While for the LA-LM pattern the impact of adding LM and LA is very similar, for the LM-LA pattern the major increase in accuracy is due to adding LA into the *None* context. This indicates that for AMEs, LA has a stronger impact on accuracy than LM, as is to be expected. The strong increase for AMEs of the LM-LA pattern when adding LA into the context is not surprising, given that the evaluators were able to see the action mentioned in the AME.

For the opposite reason, it is not surprising that AMEs show the lowest accuracy in the *None* context, given that the last action on the referent is

not seen by the evaluators. However, accuracy was still slightly over 50% in the LM-LA pattern. Overall, of the 18 instances of AMEs of the LM-LA pattern, in the *None* context a majority of evaluators correctly identified 9 and erred on the other 9. Further analysis of the difference between correctly and incorrectly identified AMEs led us to note again the important role of the mouse cursor also for AMEs.

Comparing to the LM-LA pattern, we had very low accuracy even with the *Both* context. As we mentioned in the previous section, we had very skewed test instances for AME, i.e. 18 LM-LA patterns vs. 2 LA-LM patterns. We need further investigation on the LA-LM pattern of AME with more large number of instances.

Of the 18 LM-LA instances of AMEs, there are 14 instances that mention a verb describing an action on the referent. The referents of 6 of those 14 AMEs were correctly determined by the evaluators and in all cases the mouse cursor played an important role in enabling the evaluator to determine the referent. The evaluators seem to utilize the mouse position at the time of the uttering of the referring expression as well as mouse movements in the video shown. In contrast, for 8 out of the 9 incorrectly determined AMEs no such information from the mouse was available. There was a very similar pattern for AMEs that did not include a verb. These points indicate that movements and the position of the mouse both during the video as well as the time point of the uttering of the referring expression give important clues to evaluators.

### 4.4 Other Expressions

There is a relatively even gain in identification accuracy from *None* to *Both* of between about 10–15% for both patterns. However, there is a similar tendency as for AMEs, since there is a difference between the two patterns in terms of the relative contribution of LM and LA to this increase. While for the LA-LM pattern the impact of adding LM and LA is roughly equivalent, for the LM-LA pattern the major increase in accuracy is due to adding LM into the *LA*-context.

For this pattern of OTHERs, LM has a stronger impact on accuracy than LA, which is exactly the opposite tendency to AMEs. For OTHERs (e.g. use of attributes for object identification), seeing the last action on the target has a less positive impact than listening to the last linguistic mention.

Furthermore, we note the relatively high accuracy in the *None* context for OTHERs, underlining the context-independence of expressions utilizing attributes and spatial relations of the pieces.

#### 4.5 Error Analysis

We analyzed those instances whose referents were not correctly identified by a majority of evaluators in the *Both* context. Among the three expression types, there were about 13–16% of wrong answers. In total for 7 of the 60 expressions a majority of evaluators gave wrong answers (4 DPs, 2 AMEs and 1 OTHER). Analysis of these instances indicates that some improvements of our conception of “context” is needed.

For 3 out of the 4 DPs, the mouse was not over the referent or was closer to another piece. In addition, these DPs included expressions that pointed to the role of a piece in the overall construction of the goal shape, e.g. “*soitu ga atama* (that is the head)”, or where a DP is used as part of a more complex referring expression, e.g. “*sore to onazi katati ...* (the same shape as this)”, intended to identify a different piece. For a non-participant of the task, such expressions might be difficult to understand in any context. This phenomenon is related to the “overhearer-effect” (Schober et al., 1989).

The two AMEs that the majority of evaluators failed to identify in the *Both* context were also misidentified in the *LA* context. Both AMEs were missing a verb describing an action on the referent. While for AMEs including a verb the accuracy increased from *None* to *Both* by 50%, for AMEs without a verb there was an increase by slightly over 30%, indicating that in the case where an AME lacks a verb, the context has a smaller positive impact on accuracy than for AMEs that include a verb. In order to account for those cases, further work is necessary, such as investigating how to account for the information on the distractors.

### 5 Conclusions and Future Work

In order to address the task of designing a flexible experiment set-up with relatively low cost for extrinsic evaluations of referring expressions, we investigated the context that needs to be shown to evaluators in order to correctly determine the referent of an expression.

The analysis of our results showed that the con-

text had a significant impact on referent identification. The impact was strongest for AMEs and DPs and less so for OTHERs. Interestingly, we found for both DPs and AMEs that including LA in the context had a stronger positive impact than including LM. This emphasizes the importance of taking into account extra-linguistic information in a situated domain, as considered in this study.

Our analysis of those expressions whose referent was incorrectly identified in the *Both* context indicated some directions for improving the “context” used in our experiments, for example looking further into AMEs without a verb describing an action on the referent. Generally, there is a necessity to account for mouse movements during the video shown to evaluators as well as the problem for extrinsic evaluations of how to address the “overhearer’s effect”.

While likely differing in the specifics of the set-up, the methodology in the experiment design discussed in this paper is applicable to other domains, in that it allows a low-cost flexible design of evaluating referring expressions in a dynamic domain. In order to avoid the additional effort of analyzing cases in relation to LM and LA, in the future it will be desirable to simply set a certain time period and base an evaluation on such a set-up.

However, we cannot simply assume that a longer context would yield a higher identification accuracy, given that evaluators in our set-up are not actively participating in the interaction. Thus there is a possibility that identification accuracy actually decreases with longer video segments, due to a loss of the evaluator’s concentration. Further investigation of this question is indicated.

Based on the work reported in this paper, we plan to implement an extrinsic task-performance evaluation in the dynamic domain. Even with the large potential cost-savings based on the results reported in this paper, extrinsic evaluations will remain costly. Thus one important future task for extrinsic evaluations will be to investigate the correlation between extrinsic and intrinsic evaluation metrics. This in turn will enable the use of cost-effective intrinsic evaluations whose results are strongly correlated to task-performance evaluations. This paper made an important contribution by pointing the direction for further research in extrinsic evaluations in the dynamic domain.

## References

- Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 197–200.
- Donna Byron, Thomas Mampilly, Vinay Sharma, and Tianfang Xu. 2005. Utilizing visual attention for cross-modal coreference interpretation. In *CONTEXT 2005*, pages 83–96.
- Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 165–173.
- Aoife Cahill and Josef van Genabith. 2006. Robust PCFG-based generation using automatically acquired lfg approximations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1033–1040.
- Barbara Di Eugenio, Pamela. W. Jordan, Richmond H. Thomason, and Johanna. D Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human-Computer Studies*, 53(6):1017–1076.
- Mary Ellen Foster, Ellen Gurman Bard, Markus Guhe, Robin L. Hill, Jon Oberlander, and Alois Knoll. 2008. The roles of haptic-ostensive referring expressions in cooperative, task-based human-robot dialogue. In *Proceedings of 3rd Human-Robot Interaction*, pages 295–302.
- Mary Ellen Foster, Manuel Giuliani, Amy Isard, Colin Matheson, Jon Oberlander, and Alois Knoll. 2009. Evaluating description and reference strategies in a cooperative human-robot dialogue system. In *Proceedings of the 21st international joint conference on Artificial intelligence (IJCAI 2009)*, pages 1818–1823.
- Mary Ellen Foster. 2008. Automated metrics that agree with human judgements on generated output for an embodied conversational agent. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 95–103.
- Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The give-2 corpus of giving instructions in virtual environments. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*, pages 2401–2406.
- Albert Gatt, Anja Belz, and Eric Kow. 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.
- Pamela W. Jordan and Marilyn A. Walker. 2005. Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24:157–194.
- Imtiaz Hussain Khan, Kees van Deemter, Graeme Ritchie, Albert Gatt, and Alexandra A. Cleland. 2009. A hearer-oriented evaluation of referring expression generation. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 98–101.
- Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Sara Dalzel-Job, Jon Oberlander, and Johanna Moore. 2009. Validating the web-based evaluation of nlg systems. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 301–304.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 311–318.
- Ivandr e Paraboni, Judith Masthoff, and Kees van Deemter. 2006. Overspecified reference in hierarchical domains: Measuring the benefits for readers. In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 55–62.
- Paul L.A. Piwek. 2007. Modality choice for generation of referring acts. In *Proceedings of the Workshop on Multimodal Output Generation (MOG 2007)*, pages 129–139.
- Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.
- Ehud Reiter and Somayajulu Sripada. 2002. Should corpora texts be gold standards for NLG? In *Proceedings of 2nd International Natural Language Generation Conference (INLG 2002)*, pages 97–104.
- Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58.
- Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.
- Michael F. Schober, Herbert, and H. Clark. 1989. Understanding by addressees and overhearers. *Cognitive Psychology*, 21:211–232.

Philipp Spanger, Masaaki Yasuhara, Ryu Iida, and Takenobu Tokunaga. 2009a. A Japanese corpus of referring expressions used in a situated collaboration task. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 110 – 113.

Philipp Spanger, Masaaki Yasuhara, Iida Ryu, and Tokunaga Takenobu. 2009b. Using extra linguistic information for generating demonstrative pronouns in a situated collaboration task. In *Proceedings of PreCogSci 2009: Production of Referring Expressions: Bridging the gap between computational and empirical approaches to reference*.

Karen Sparck Jones and Julia R. Galliers. 1996. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Linguistics and Intelligent Text Processing*, pages 341–351. Springer-Verlag.

Laura Stoia, Darla Magdalene Shockley, Donna K. Byron, and Eric Fosler-Lussier. 2006. Noun phrase generation for situated dialogs. In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 81–88.

Kees van Deemter. 2007. TUNA: Towards a unified algorithm for the generation of referring expressions. Technical report, Aberdeen University. [www.csd.abdn.ac.uk/research/tuna/pubs/TUNA-final-report.pdf](http://www.csd.abdn.ac.uk/research/tuna/pubs/TUNA-final-report.pdf).

Ielka van der Sluis, Albert Gatt, and Kees van Deemter. 2007. Evaluating algorithms for the generation of referring expressions: Going beyond toy domains. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*.



# Harvesting Re-usable High-level Rules for Expository Dialogue Generation

**Svetlana Stoyanchev**

Centre for Research in Computing  
The Open University  
Walton Hall, Milton Keynes, UK  
s.stoyanchev@open.ac.uk

**Paul Piwek**

Centre for Research in Computing  
The Open University  
Walton Hall, Milton Keynes, UK  
p.piwek@open.ac.uk

## Abstract

This paper proposes a method for extracting high-level rules for expository dialogue generation. The rules are extracted from dialogues that have been authored by expert dialogue writers. We examine the rules that can be extracted by this method, focusing on whether different dialogues and authors exhibit different dialogue styles.

## 1 Introduction

In the past decade, a new area of Natural Language Generation (NLG) has emerged: the automated generation of expository dialogue, also often referred to as scripted, authored or fictive dialogue. Research in this area began with the seminal study by André et al. (2000), which explored generation of dialogues between a virtual car buyer and seller from technical data on a car. This strand of work was developed further in the NECA project (van Deemter et al., 2008) and has since been extended to other domains, including explanation of medical histories (Williams et al., 2007), patient information leaflets (Piwek et al., 2007) and Wall Street Journal articles (Hernault et al., 2008).

Systems for generating expository dialogue have explored different inputs (databases, knowledge representations and text), generation methods (e.g., rule versus constraint-based approaches) and outputs (from dialogue scripts in text form to audio and computer-animated dialogue). A common trait of all these systems is, however, that at some point in the generation process, they produce a dialogue script, a

text file which specifies what the interlocutors say, possibly enriched with mark-up for dialogue acts, speech and gestures – see, e.g., Piwek et al. (2002). These systems are different from conventional dialogue systems in that the system does not engage in a dialogue with the user; rather, the system generates a dialogue between two or more fictitious characters for the user/audience to view and learn from. In other words, the dialogue is used to deliver information to the user or audience, rather than between the interlocutors. Piwek (2008) discusses several empirical studies that identify benefits of the use of expository dialogue for education and persuasion.

In this paper, we take a step towards addressing two shortcomings of the work so far. Firstly, all the work cited has relied on hand-crafted resources (typically rules) for creating the dialogue. With the resources being created by non-expert dialogue authors (e.g., academic researchers), generated dialogues based on these resources may not be optimal; for instance, Williams et al. (2007) found that generated dialogues can be too information-dense, requiring conversational padding. Secondly, the resources for creating dialogue are tied to a specific domain, making it hard to redeploy a system in new domains.

We propose to address the first issue by automatically creating dialogue generation resources from a corpus of dialogues written by known effective dialogue authors. This fits in with a trend in dialogue modelling and generation to create resources from empirical data (Oh and Rudnicky, 2002; DeVault et al., 2008; Henderson et al., 2008; Belz and Kow, 2009).

The second issue is addressed by specifying di-

dialogue generation rules at a level of detail that abstracts over the particulars of the domain and fits in with existing NLG architectures. The reference architecture of Reiter and Dale (2000) identifies three principal NLG tasks: Document Planning (DP), Microplanning and Realisation. DP is primarily non-linguistic: it concerns selection of information and organization of this information into a coherent whole. The latter is achieved by making sure that the information is tied together by Rhetorical Relations such as Contrast, Elaboration and Explanation, in other words, it is part of a Rhetorical Structure. We propose that dialogue generation rules interface with Rhetorical Structure and map to a Sequence of Dialogue Acts.

Interestingly, the interface between DP and Microplanning has also been identified as a place where decisions and preferences regarding style take an effect (McDonald and Pustejovsky, 1985). A question that we explore in this paper is whether dialogue styles exist at the highly abstract level we focus on in this paper. We concentrate on style in the sense of '[t]he manner of expression characteristic of a particular writer'<sup>1</sup>.

The remainder of this paper is set up as follows. In Section 2, we introduce the corpus that we use to extract dialogue generation resources. Section 3 examines the dialogues in the corpus for prima facie evidence for stylistic differences between authors at the dialogue level. In Section 4, we describe our approach to extracting high-level dialogue generation rules from the corpus. Next, in Section 5 we analyse the resulting rules, looking for further evidence of different dialogue styles. We also compare the rules that were harvested from our corpus with hand-crafted rules in terms of content and variety. Finally, Section 6 contains our conclusions and a discussion of avenues for further research.

## 2 A Parallel Monologue-Dialogue Corpus

The current work makes use of a corpus of human-authored dialogues, the CODA corpus.<sup>2</sup> In total, this corpus consist of about 800 dialogue turns. This

<sup>1</sup>From definition 13.a. of the Oxford English Dictionary at <http://dictionary.oed.com>

<sup>2</sup>Further information on the construction of this corpus can be found in the annotation manual at [computing.open.ac.uk/coda/AnnotationManual.pdf](http://computing.open.ac.uk/coda/AnnotationManual.pdf).

paper is based on three dialogues from the corpus: George Berkeley's 'Dialogues between Hylas and Philonous' (extract of 172 turns), Mark Twain's 'What is man?' (extract of 445 turns) and Yuri Gurevich's 'Evolving Algebras' (extract of 89 turns). Berkeley's dialogue is one of the classics of philosophy, arguing for the, at first sight, extravagant claim that 'there is no such thing as *material substance* in the world'. Twain, according to the Encyclopaedia Britannica 'one of America's best and most beloved writers', takes on the concept of free will. Gurevich's dialogue deals with the mathematical concept of evolving algebras. Of these dialogues, Twain is by a large margin the longest (over 800 turns in total) and the only one which is aimed specifically at the general public, rather than an academic/specialist audience.

For each of the dialogues, the corpus also contains human-authored monologue which expresses the same content as the dialogue. Monologue and dialogue are aligned through mappings from monologue snippets to dialogue spans. As a result, the CODA corpus is a parallel monologue-dialogue corpus. Both the monologue and dialogue come with annotations: the monologue with Rhetorical Structure Theory (RST) relations (Mann and Thompson, 1988; Carlson and Marcu, 2001) and the dialogue side with an adaptation of existing Dialogue Act annotation schemes (Carletta et al., 1997; Core and Allen, 1997). Table 2 contains an overview of these RST relations and Dialogue Act labels.

## 3 Dialogue Analysis

In this section we examine whether there is prima facie evidence for differences in style between the three dialogues. Whereas existing work in NLG on style has focused on lexical and syntactic choice, see Reiter and Williams (2008), here we focus on higher-level characteristics of the dialogues, in particular, proportion of turns with multiple dialogue acts, frequencies of dialogue act bigrams, and relation between dialogue acts and speaker roles.

An important reason for determining whether there are different styles involved, is that this has implications for how we use the corpus to create expository dialogue generation resources. If different dialogues employ different styles, we need to be

RST relations	Dialogue Acts
Enablement, Cause, Evaluation ( <i>Subjective, Inferred</i> ), Comment, Attribution, Condition-Hypothetical, Contrast, Comparison, Summary, Manner-means, Topic-Comment ( <i>Problem-Solution, Statement-Response, Question-Answer, Rhetorical Question</i> ) Background, Temporal, Elaboration/Explanation, ( <i>Additional, General-Specific, Example, Object-attribute, Definition, Evidence, Reason</i> ), Same-unit	Explain, Info-Request ( <i>Init-Factoid-InfoReq, Init-YN-InfoReq, Init-Complex-InfReq</i> ), Init-Request-Clarify, Response-Answer ( <i>Resp-Answer-Yes/No, and Resp-Answer-Factoid</i> ), Resp-Agree, Resp-Contradict

Table 1: RST relations and Dialogue Acts used in the CODA corpus. Annotators used the fine-grained categories in italics that are listed in brackets. For the current study, we rely, however, on the higher-level categories that precede the fine-grained categories and which combine several of them.

careful with creating resources which combine data from different dialogues. Merging such data, if anything, may lead to the generation of dialogues which exhibit features from several possibly incompatible styles. Since our aim is specifically to generate dialogues that emulate the masters of dialogue authoring, it is then probably better to create resources based on data from a single master or dialogue.

### 3.1 Multi-act Turns

One of the characteristics of dialogue is the pace and the amount of information presented in each of the speaker’s turns. In a fast-paced dialogue turns are concise containing a single dialogue act. Such dialogues of the form *A:Init B:Response A:Init B:Response ...* are known as ‘pingpong’ dialogue. Twain’s ‘What is man?’ dialogue starts in this fashion (O.M. = Old Man; Y.M = Young Man):

- O.M. What are the materials of which a steam-engine is made?  
 Y.M. Iron, steel, brass, white-metal, and so on.  
 O.M. Where are these found?  
 Y.M. In the rocks.  
 O.M. In a pure state?  
 Y.M. No—in ores.  
 ...

One character serves as the initiator and the other replies with a response. With turns that contain more than one dialogue, henceforth multi-act turns, this pattern can be broken:

- O.M. ...  
 And you not only did not make that

Author	Twain	Gurevich	Berkeley
Multi-act	34%	43%	24%
Layman/Expert	45%/55%	36%/64%	51%/49%

Table 2: Proportion of multi-act utterances and their distribution between Layman and Expert

- machinery yourself, but you have NOT EVEN ANY COMMAND OVER IT.  
 Y.M. This is too much.  
 You think I could have formed no opinion but that one?  
 O.M. Spontaneously? No. And ...

Multi-act turns are turns comprised of multiple dialogue acts, such as the Young Man’s in the example above, where a Resp-Contradict (‘This is too much.’) is followed by an Init-YN-Request (‘You think I could have formed no opinion but that one?’).

The dialogue pace may vary throughout a dialogue. We, however, find that overall proportions of multi-act turns and their distribution between expert and layman vary between the authors (see Table 2). Gurevich’s dialogue has the highest proportion (43%) of multi-act turns and majority of them are attributed to the *expert*. Only 24% of Berkeley’s dialogue turns consist of multiple dialogue acts and they are evenly split between the expert and the layman. Gurevich’s dialogue is the type of dialogue where an expert gives a lesson to a layman while in Berkeley’s dialogue one character often complements ideas of the other character making it difficult to determine which of the characters is an *expert*. The amount of multi-act turns seems to be one of the stylistic choices made by a dialogue author.

### 3.2 Dialogue Diversity

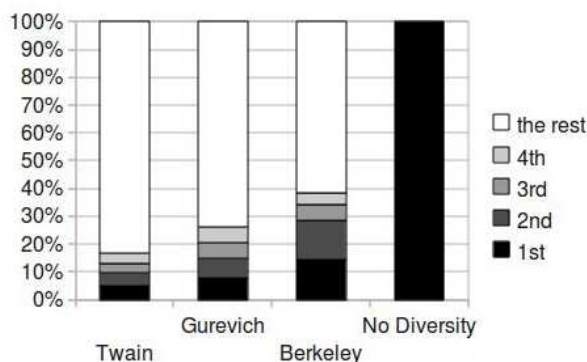


Figure 1: *Bigram coverage* for the 1-st to 4th most frequent bigrams.

Dialogues are essentially a sequence of turns, where each turn consists of one or more dialogue acts. For our measure of dialogue diversity we focus on two-turn sequences (i.e., turn bigrams), where a turn is identified by the sequence of dialogue acts it contains.

We define *bigram coverage for  $i$*  as the percentage that the top  $i$  most frequent bigrams contribute to all bigrams in the corpus. Diversity of the dialogue is inversely related to the *dialogue coverage*. In a dialogue with minimal diversity, the same turn, consisting of one or more dialogue acts, is repeated throughout the dialogue. The turn bigram consisting of two such turns has 100% bigram coverage.

Figure 1 shows the coverage for  $1 \leq i \leq 4$  for each author in the corpus.<sup>3</sup> Out of the three authors, Twain's dialogues are the most diverse where the top 4 bigrams constitute only 15% of all bigrams. In Gurevich's dialogues the four most frequent bigrams constitute 25% and in Berkeley 40%.

Note that for all three authors the dialogue coverage for the 4 most frequent bigrams is quite low indicating high variability in bigrams used. To achieve such variability in automatically generated dialogues we need a large number of distinct generation rules.

<sup>3</sup>This range was chosen for illustration purposes. *Bigram coverage* can be compared for any  $i \leq \text{total number of distinct bigrams}$ .

### 3.3 Dialogue Acts and Speaker Roles

One of the most frequent bigrams for all three authors was, not unexpectedly, the sequence:

A: InfoRequest  
B: Response-Answer

There is, however, a difference in the roles of speakers A and B. In all dialogues, one of the speakers took on the expert role and the other the layman role. For the aforementioned bigram, both in Berkeley's and Gurevich's dialogues the layman typically initiates the request for information and the expert responds (and often goes on to explain the response in Gurevich's dialogue):

Q: Is it difficult to define basic transition rules in full generality?  
A: No. Here is the definition.  
– Any local function update is a rule.

...  
(From Gurevich's dialogue)

In contrast, in Twain's dialogues the roles are typically reversed: the expert asks and the layman responds:

O.M. Then the impulse which moves you to submit to the tax is not ALL compassion, charity, benevolence?  
Y.M. Well—perhaps not.

Both techniques allow the author to convey a particular piece of information, but each giving rise its very own dialogue style.

## 4 Approach to Rule Extraction

Comparing statistics for individual dialogues gives us some idea about whether different styles are involved. The true test for whether different styles are involved is, however, whether for the *same* content different realizations are generated. Unfortunately, for our three dialogues the content is different to begin with. The parallel corpus allows us, however, to get around this problem. From the parallel corpus we can extract rules which map RST structures to dialogue act sequences. The Lefthand Side (LHS) of a rule represents a particular rhetorical structure found in the monologue side, whereas the Righthand Side (RHS) of the rule represents the dialogue act sequence with which it is aligned in the corpus.

Such rules can be compared between the different dialogues: in particular, we can examine whether the *same* LHS gives rise to similar or different RHSs.

#### 4.1 Comparison with previous work

Hernault et al. (2008) manually construct surface-level rules mapping monologue to dialogue. Surface-level rules execute text-to-text conversion operating directly on the input string. In our approach, we separate the conversion into two stages. A first stage converts RST structures to Dialogue Act sequences. A second stage, which is beyond the scope of this paper, converts Dialogue Act sequences to text.

A further difference between the current approach and Hernault et al.’s is that the LHS of our rules can match nested RST structures. This covers, what we call, *simple rules* (involving a single RST relation, e.g., Contrast(X,Y)) and *complex rules* (involving 2 or more nested RST relations, e.g., Contrast(Condition(X,Y),Z)). Hernault et al. only allow for simple rules. A detailed comparison between our approach and that of Hernault et al., using the attribution rule as an example, can be found in Section 5.3.

id	DA	turns
0	Init-YN-InfoReq	Is your mind a part of your PHYSICAL equipment ?
0	Resp-Answer-No	No.
1	Explain	It is independent of it ; it is spiritual
2	Init-YN-InfoReq	Being spiritual, it cannot be affected by physical influences?
2	Resp-Answer-No	No.
3	Init-YN-InfoReq	Does the mind remain sober with the body is drunk ?
-	decorative	Well-
3	Resp-Answer-No	No.

Table 3: Example of annotated dialogue (from Mark Twain’s ‘What is man?’).

#### 4.2 Rule Extraction Algorithm

Table 3 and Figure 2 show annotated dialogue (authored by Twain) and its annotated monologue translation. Each terminal node of the RST structure corresponds to a part of a monologue snippet. All nodes with the same *id* correspond to a complete

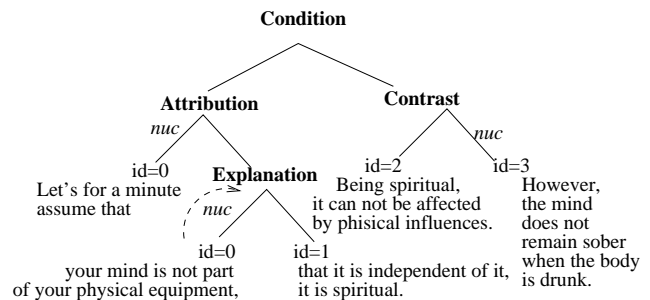


Figure 2: RST structure for the translation of dialogue in Table 3

span	rule
0-0	Attribution(0, 0)
0-1	Attribution( Explanation(0, 1))
2-3	Contrast(2, 3)
0-3	Condition (Attribution( Explain(0, 1)), Contrast(2, 3))

Table 4: RST sub-structures: LHS of monologue-to-dialogue mapping rules

snippet and are linked to the dialogue act(s) with the same *ids*. The relation between monologue snippets and dialogue act segments is one-to-many. In other words, one snippet (e.g. snippets with *id=0*, *id=2*) can be expressed by multiple dialogue act segments.

Rules are extracted as follows: For each (automatically extracted) sub-structure of the RST structures on the monologue side, a rule is created (see Table 4). Two constraints restrict extraction of sub-structures: 1) spans of the structure’s terminal nodes must be consecutive and 2) none of the *ids* of the terminal nodes are shared with a node outside the sub-structure.

For example, *Explanation(0, 1)* is not extracted because the node with *id=0* appears also under the *Attribution* relation which is not a part of this sub-structure.

Additionally, rules are generated by removing a relation and its satellite node and moving a nucleus node one level up. *Attribution(0, 0)* was extracted from a tree that had the *Explanation* relation and its satellite child *I* pruned. This operation relies on the validity of the following principle for RST (Marcu, 1997): ‘If a relation holds between two textual spans of the tree structure of a text, that relation also holds between the most important units of the constituent

subspans.’

The RST sub-structure is the LHS of a rule and dialogue act sequences are the RHS of a rule.

## 5 Results: Analysis of the Rules

In this section we describe the rules collected from the corpus. We compare the rules collected from the dialogues of different authors. We also compare the rules constructed manually in previous work with the rules collected from the corpus, specifically for the *attribution* relation.

### 5.1 Rule Statistics

relation	Twain	Gurev	Berk	all
simple	31 (33)	29 (38)	25 (26)	<b>81</b> (97)
complex	19	26	16	<b>61</b> (61)
null	15 (22)	9 (18)	9 (27)	<b>25</b> (67)
total	65	64	50	<b>167</b>
# turns	85	78	96	<b>259</b>

Table 5: Numbers of extracted distinct structural rules (total occurrences are parenthesized)

relation	Twain	Gurevich	Berkley
attribution	15%	2%	12%
contrast	18%	9%	17%
expl/elab	34%	47%	26%
eval	9%	6%	21%
other	24%	36%	24%
total	100%	100%	100%

Table 6: Proportions of relations expressed as rules

relation	Twain	Gurevich	Berkley
overall	2.4	1.9	2.9
contrast	2.3	2	2.6
elab/expl	2.7	1.7	3.3
eval	2	2	2.5

Table 7: Average number of turns in simple rules

*Simple* rules are the rules with one RST relation in the LHS. *Complex* rules are the rules with multiple RST relations in the LHS. In Table 4, rules for the LHS 0-0 and 2-3 are simple while the rules for 0-1

and 0-3 are complex. *Null* rules are the rules with no RST relation in the LHS.

From our sample of 259 translated and annotated dialogue turns from the corpus, we extracted 81 simple, 61 complex, and 25 null rules (null rules involve no RST structure and are discussed below). Table 5 shows the number of distinct rules per author.<sup>4</sup> In parentheses we show the number of actual (not necessarily distinct) rule occurrences in corpus. The majority of *simple* rules in the corpus (65 out of 81) occur only once.<sup>5</sup> This shows that the dialogue authors use a variety of dialogue act sequences when presenting their arguments in dialogue.

To compare dialogue *styles* we compare the rules across the dialogues of different authors. Table 6 shows the proportions of relation types in each author’s dialogues that are mapped to a dialogue structure and produce a mapping rule.<sup>6</sup> Not all relations in monologue are mapped to a dialogue structure. For example, *Explain* moves may contain multiple clauses that are presented by a single character in the same turn. We find differences in distributions of relation types mapped to dialogue between the three authors (Fisher’s exact test  $p < .01$ ). Berkeley’s dialogues produce more mapping rules with *Evaluation* and less with *Explanation/Elaboration* relations than the other two authors. Gurevich’s dialogues produce less mapping rules with *Attribution* and *Contrast* relations than the other two authors. This difference between distributions of relation types mapped to dialogue has an important implication for dialogue generation. Dialogue generation programs may vary the style of a dialogue by choosing which discourse relations of the monologue are mapped to dialogue turns.

Another relevant property of a rule is the number of turns in the RHS of the rule. Number of turns in a rule shows how many times the dialogue characters switch to present information of the monologue corresponding to the LHS of the rule. The average numbers of turns in the RHS of all rules of the Twain, Gurevich, and Berkeley dialogues are 2.4, 1.9, 2.9 respectively (see Table 7). They are all pairwise significantly different (t-test  $p < .05$ ) ranking the au-

<sup>4</sup>Two rules are distinct if either their LHS (relation in monologue) or RHSs (sequence of dialogue acts) are different.

<sup>5</sup>65=81-(97-81)

<sup>6</sup>This includes simple and complex rules

thors in the order *Gurevich* < *Twain* < *Berkeley* according to the number of turns in the RHS of the rule. Similar ranking also appears as a trend for individual relations suggesting that this is the effect of the author's style rather than the relations (the distribution of relation types is different across the authors). This suggests that dialogue generation may affect the style of automatically generated dialogue by selectively choosing rules with longer (or shorter) RHS.

## 5.2 Null Rule

A *null* rule is a rule where a sequence of dialogue turns between two characters corresponds with a text segment with no rhetorical relation. A text segment without a rhetorical relation corresponds to a leaf node in the RST structure. A *null rule* typically creates a dialogue fragment consisting of a yes/no question (*Init-YN-Info-Req*) followed by yes/no answer, or a complex information request (e.g. What is your opinion on X?) followed by an *Explain* dialogue act, or a presentation of an argument (*Explain* dialogue act) followed by a response that signals agreement (*Resp-Agree*). *Null* rules create more interactivity in the dialogue.

The monologue segment corresponding to the LHS of a *null* rule may be in a rhetorical relation with another segment, such that the LHS of the *null* rule is embedded into another rule. Table 8 shows an example of a *null rule* embedded in a *contrast rule*. Turns 1 - 3 correspond to the RHS of the *Null* rule and 1 - 4 correspond to the RHS of the *Contrast* rule.

*Null* rules can be used to turn information into dialogue, even when there is no RST relation. For example, we may want to convey the piece of information A,B,C,D,E in that order, with *rel1*(A,B) and *rel2*(D,E). Whereas a simple rule may apply to relations and turn them into dialogue, C is left untouched. However, a *null* rule can be applied to C, to also turn its presentation into a dialogue exchange.

## 5.3 Case Study: the Attribution Rule

In this section we present a comparison of manually created rules for the RST attribution relation and rules extracted from the CODA corpus.

Hernault et al. manually construct two surface-level rules for the **Attribution (S,N)**<sup>7</sup> relation (see

<sup>7</sup>N is a nucleus phrase that carries main information and S is

Table 9). In the *Dialogue Act* column we show the dialogue act representation of the corresponding surface-level rules. The first rule converts attribution relation into a *Complex-Info-Request* by the Layman followed with the *Explain* by the Expert. The second rule converts the attribution relation into *Explain* by the Expert, *Factoid-Info-Request* by the Layman and *Factoid-Response* by Expert. In both rules, the Expert is the one providing information (N) to the Layman and information is presented in *Explain* dialogue act

Table 10 shows six attribution rules we collected from phrases with attribution relation in the corpus (Twain1-4,Berkeley1,Gurevich)<sup>8</sup>. We notice several differences with the manually constructed rules:

- The variety of dialogue act sequences: each RHS of the rule (or dialogue act sequence) is different.
- Main information (N) can be presented by either the expert (Twain1, Twain2, Twain3, Berkeley1) or by the layman (Twain4, Gurevich1).
- Main information (N) can be presented in different dialogue acts: *Explain* dialogue act (Twain1, Twain4, Berkeley), *YN-Info-Request* (Twain2, Twain3), or *Complex-Info-Request* (Gurevich).
- Contextual information is part of the rule and may be used when choosing which rule to apply.

## 6 Conclusions and Further Work

In this paper, we have introduced a new approach to creating resources for automatically generating expository dialogue. The approach is based on extracting high-level rules from RST relations to Dialogue Act sequences using a parallel Monologue-Dialogue corpus. The approach results in rules that are reusable across applications and based on known expert dialogue authors.

After examining differences between the dialogues in the corpus in order to obtain prima facie evidence for differences in style, we conducted a detailed evaluation of the rules that were extracted

a satellite phrase that contains the entity to whom N is attributed

<sup>8</sup>These are all the rules for attribution RST relation from 50 annotated turns for each author

Turn	Speaker	Dialogue act	Dialogue
<b>Contrast rule.</b> Segment with contrast relation: [He never does anything for any one else's comfort , spiritual or physical.] [EXCEPT ON THOSE DISTINCT TERMS – that it shall FIRST secure HIS OWN spiritual comfort ].			
<b>Null rule.</b> Segment without rhetorical relation: He never does anything for any one else's comfort , spiritual or physical			
1	Layman	decorative	Come!
2	Expert	Init-YN-Request	He never does anything for any one else ' s comfort , spiritual or physical ?
3	Expert	Resp-Answer-No	No
4	Expert	Explain	EXCEPT ON THOSE DISTINCT TERMS – that it shall FIRST secure HIS OWN spiritual comfort .

Table 8: **Contrast rule** example containing **null rule** from Twain dialogue.

Rule 1			
Speaker	Surface-level Rule	Dialogue act	Example Dialogue
Layman	What did + <i>GetSubject(S+N)</i> + <i>GetMain-VerbLemma(S+N)</i>	Complex-Info-Request	What did S say?
Expert	<i>AddifNotPresentIn(N, That)</i> + N	Explain	N
Rule 2			
Expert	<i>RemoveIfPresentIn(N, That)</i> + N	Explain	N
Layman	Who <i>GetMainVerb(N)</i> that?	Factoid-Info-Req	Who said that?
Expert	<i>GetSubjectFromSentence(S+N)</i>	Factoid-Response	S did

Table 9: Manually created rules for **Attribution(S,N)** relation (Hernault et al., 2008)

from the corpus. We extracted 167 distinct rules and discussed the three types of rules: null, simple and complex (depending on the number of RST relation in the LHS: 0, 1 or more).

We found differences between authors in several respects, specifically:

- number of turns per simple rule
- number of dialogue acts per simple rule
- combination of speaker roles and dialogue acts

A detailed comparison between our automatically extracted attribution rule and the hand-crafted rules used by Hernault et al. showed up a number of differences. Apart from the fact that the corpus yielded many more rules than the two manually created ones, there were differences in which interlocutor presented particular information and which dialogue acts were being used.

The current work has focussed on high-level mapping rules which can be used both for generation from databases and knowledge representations and also for generation from text. In future work, we will focus on mapping text (in monologue form) to dialogue. For this we need to combine the high-level rules with rules for paraphrasing the text in the monologue with text for the dialogue acts that express the same information in dialogue form. For

automatically extracting these surface level mappings we will draw on the approach to learning paraphrases from a corpus that is described in Barzilay and McKeown (2001). An important component of our future effort will be to evaluate whether automatically generating dialogues from naturally-occurring monologues, following the approach described here, results in dialogues that are fluent and coherent and preserve the information from the input monologue.

## Acknowledgements

We would like to thank the anonymous reviewers of INLG2010 for their helpful comments and our colleagues in the Open University's NLG group for stimulating discussions on the content of this paper. The research reported in this paper was carried out as part of the CODA project (Coherent Dialogue Automatically generated from text; see <http://computing.open.ac.uk/coda/>) which is funded by the UK Engineering and Physical Sciences Research Council under grant EP/G/020981/1.

## References

- E. André, T. Rist, S. van Mulken, M. Klesen, and S. Baldes. 2000. The automated design of believable dialogues for animated presentation teams. In *Em-*



Speaker	Dialogue act	Dialogue
<b>Twain1</b> <i>I will put that law into words, keep it in your mind:</i> FROM HIS CRADLE TO HIS GRAVE A MAN NEVER DOES... <b>Satellite of Summary</b>		
Layman	Init-YN-InfoReq	Will you put that law into words?
Expert	Resp-Answer-Yes	Yes.
Expert	Resp-Explain	This is the law, keep it in your mind. FROM HIS CRADLE TO HIS GRAVE A MAN NEVER DOES...
<b>Twain2</b> <i>I can not imagine that there is some other way of looking at it.</i> <b>Satellite of Explanation</b>		
Expert	Init-Complex-InfoReq /clarify	What makes you think that?
Layman	<i>decorative</i>	Pray what else could I think?
Expert	Init-YN-InfoReq	Do you imagine that there is some other way of looking at it?
<b>Twain3</b> <i>One cannot doubt that he felt well.</i> <b>Satellite of Evaluation-Conclusion</b>		
Expert	Init-YN-InfoReq	He felt well?
Layman	Resp-Answer-Yes	One cannot doubt it.
<b>Twain4</b> <i>As I said a minute ago</i> Hamilton fought that duel to get PUBLIC approval. <b>Nucleus of Explanation</b>		
Layman	Init-Explain/contradict	A minute ago you said Hamilton fought that duel to get PUBLIC approval.
Resp-Agree	Resp-Agree	I did.
<b>Berkeley1</b> <i>You can not conceive a vehement sensation to be without pain or pleasure.</i>		
Expert	Init-Explain	Again, try in your thoughts, Hylas, if you can conceive a vehement sensation to be without pain or pleasure.
Layman	Resp-Contradict	You can not.
<b>Gurevich</b> <i>I will explain what static algebras are exactly.</i> <b>Nucleus of Statement-response</b>		
Layman	Init-Complex-InfoReq	Please explain to me what static algebras are exactly.
Expert	Resp-Agree	Gladly.

Table 10: Attribution Examples. Satellite is *italicised*.

- bodied Conversational Agents*, pages 220–255. MIT Press, Cambridge, Mass.
- R. Barzilay and K. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the ACL*, Toulouse, France.
- A. Belz and E. Kow. 2009. System Building Cost vs. Output Quality in Data-to-Text Generation. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG'09)*, Athens, Greece.
- J. Carletta, A. Isard, and J. C. Kowtko. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23:13–31.
- L. Carlson and D. Marcu. 2001. Discourse tagging reference manual. Technical Report ISI-TR-545, ISI, September.
- M. Core and J. Allen. 1997. Coding dialogs with the damsl annotation scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machine*.
- D. DeVault, D. Traum, and R. Artstein. 2008. Making Grammar-Based Generation Easier to Deploy in Dialogue Systems. In *Procs SIGdial 2008*, Ohio, June.
- E. Reiter and S. Williams. 2008. Three approaches to generating texts in different styles. In *Proceedings of the Symposium on Style in text: creative generation and identification of authorship*.
- J. Henderson, O. Lemon, and K. Georgila. 2008. Hybrid Reinforcement / Supervised Learning of Dialogue Policies from Fixed Datasets. *Computational Linguistics*, 34(4):487–511.
- H. Hernault, P. Piwek, H. Prendinger, and M. Ishizuka. 2008. Generating dialogues for virtual agents using nested textual coherence relations. In *IVA08: 8th International Conference on Intelligent Virtual Agents*.
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.
- D. Marcu. 1997. From Discourse Structures to Text Summaries. In *The Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain.
- D. McDonald and J. Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Proceedings of the second conference on European chapter of the Association for Computational Linguistics*, pages 187–193, Geneva, Switzerland.
- A. Oh and A. Rudnicky. 2002. Stochastic natural language generation for spoken dialog. *Computer Speech and Language*, 16(3/4):387–407.
- P. Piwek, B. Krenn, M. Schroeder, M. Grice, S. Baumann, and H. Pirker. 2002. RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA. In *Proceedings of the AAMAS workshop "Embodied conversational agents - let's specify and evaluate them!"*, Bologna, Italy, July.
- P. Piwek, H. Hernault, H. Prendinger, and M. Ishizuka. 2007. T2D: Generating Dialogues between Virtual Agents Automatically from Text. In *Intelligent Virtual Agents*, LNAI 4722, pages 161–174. Springer Verlag.
- P. Piwek. 2008. Presenting Arguments as Fictive Dialogue. In *Proceedings of 8th Workshop on Computa-*

- tional Models of Natural Argument (CMNA08)*, Patras, Greece, July. ISBN 978-960-6843-12-9.
- E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.
- K. van Deemter, B. Krenn, P. Piwek, M. Klesen, M. Schroeder, and S. Baumann. 2008. Fully generated scripted dialogue for embodied agents. *Artificial Intelligence Journal*, 172(10):1219–1244.
- S. Williams, P. Piwek, and R. Power. 2007. Generating Monologue and Dialogue to Present Personalised Medical Information to Patients. In *Procs ENLG 2007*, pages 167–170, Schloss Dagstuhl, Germany.

# Feature Selection for Fluency Ranking

Daniël de Kok

University of Groningen  
d.j.a.de.kok@rug.nl

## Abstract

Fluency rankers are used in modern sentence generation systems to pick sentences that are not just grammatical, but also fluent. It has been shown that feature-based models, such as maximum entropy models, work well for this task.

Since maximum entropy models allow for incorporation of arbitrary real-valued features, it is often attractive to create very general feature templates, that create a huge number of features. To select the most discriminative features, feature selection can be applied. In this paper we compare three feature selection methods: frequency-based selection, a generalization of maximum entropy feature selection for ranking tasks with real-valued features, and a new selection method based on feature value correlation. We show that the often-used frequency-based selection performs badly compared to maximum entropy feature selection, and that models with a few hundred well-picked features are competitive to models with no feature selection applied. In the experiments described in this paper, we compressed a model of approximately 490.000 features to 1.000 features.

## 1 Introduction

As shown previously, maximum entropy models have proven to be viable for fluency ranking (Nakanishi et al., 2005; Velldal and Oepen, 2006; Velldal, 2008). The basic principle of maximum entropy models is to minimize assumptions, while imposing constraints such that the expected feature value

is equal to the observed feature value in the training data. In its canonical form, the probability of a certain event ( $y$ ) occurring in the context ( $x$ ) is a log-linear combination of features and feature weights, where  $Z(x)$  is a normalization over all events in context  $x$  (Berger et al., 1996):

$$p(y|x) = \frac{1}{Z(x)} \exp \sum_{i=1}^n \lambda_i f_i \quad (1)$$

The training process estimates optimal feature weights, given the constraints and the principle of maximum entropy. In fluency ranking the input (e.g. a dependency structure) is a context, and a realization of that input is an event within that context.

Features can be hand-crafted or generated automatically using very general feature templates. For example, if we apply a template *rule* that enumerates the rules used to construct a derivation tree to the partial tree in figure 1 the *rule(max\_xp(np))* and *rule(np\_det\_n)* features will be created.

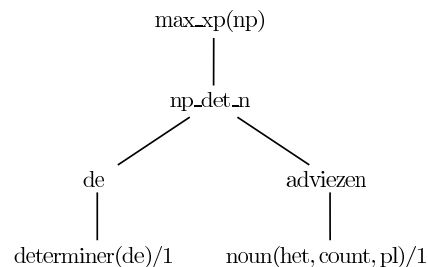


Figure 1: Partial derivation tree for the noun phrase *de adviezen* (*the advices*).

To achieve high accuracy in fluency ranking quickly, it is attractive to capture as much of the lan-

guage generation process as possible. For instance, in sentence realization, one could extract nearly every aspect of a derivation tree as a feature using very general templates. This path is followed in recent work, such as Velldal (2008). The advantage of this approach is that it requires little human labor, and generally gives good ranking performance. However, the generality of templates leads to huge models in terms of number of features. For instance, the model that we will discuss contains about 490,000 features when no feature selection is applied. Such models are very opaque, giving very little understanding of good discriminators for fluency ranking, and the size of the models may also be inconvenient. To make such models more compact and transparent, feature selection can be applied.

In this paper we make the following contributions: we modify a maximum entropy feature selection method for ranking tasks; we introduce a new feature selection method based on statistical correlation of features; we compare the performance of the preceding feature selection methods, plus a commonly used frequency-based method; and we give an analysis of the most effective features for fluency ranking.

## 2 Feature Selection

### 2.1 Introduction

Feature selection is a process that tries to extract  $S \subset F$  from a set of features  $F$ , such that the model using  $S$  performs comparably to the model using  $F$ . Such a compression of a feature set can be obtained if there are features: that occur sporadically; that correlate strongly with other features (features that show the same behavior within events and contexts); or have values with little or no correlation to the classification or ranking.

Features that do have no correlation to the classification can be removed from the model. For a set of highly-correlating features, one feature can be selected to represent the whole group.

Initially it may seem attractive to perform fluency selection by training a model on all features, selecting features with relatively high weights. However, if features overlap, weight mass will usually be divided over these features. For instance, suppose that  $f_1$  alone has a weight of 0.5 in a given model. If

we retrain the model, after adding the features  $f_2..f_5$  that behave identically to  $f_1$ , the weight may be distributed evenly between  $f_1..f_5$ , giving each feature the weight 0.1.

In the following sections, we will give a short overview of previous research in feature selection, and will then proceed to give a more detailed description of three feature selection methods.

### 2.2 Background

Feature selection can be seen as model selection, where the best model of all models that can be formed using a set of features should be selected. Madigan and Raftery (1994) propose an method for model selection aptly named *Occam's window*. This method excludes models that do not perform competitively to other models or that do not perform better than one of its submodels. Although this method is conceptually firm, it is nearly infeasible to apply it with the number of features used in fluency ranking. Berger et al. (1996) propose a selection method that iteratively builds a maximum entropy model, adding features that improve the model. We modify this method for ranking tasks in section 2.5. Ratnaparkhi (1999) uses a simple frequency-based cutoff, where features that occur infrequently are excluded. We discuss a variant of this selection criterium in section 2.3. Perkins et al. (2003) describe an approach where feature selection is applied as a part of model parameter estimation. They rely on the fact that  $\ell_1$  regularizers have a tendency to force a subset of weights to zero. However, such integrated approaches rely on parameter tuning to get the requested number of features.

In the fluency ranking literature, the use of a frequency cut-off (Velldal and Oepen, 2006) and  $\ell_1$  regularization (Cahill et al., 2007) is prevalent. We are not aware of any detailed studies that compare feature selection methods for fluency ranking.

### 2.3 Frequency-based Selection

In frequency-based selection we follow Malouf and Van Noord (2004), and count for each feature  $f$  the number of inputs where there are at least two realizations  $y_1, y_2$ , such that  $f(y_1) \neq f(y_2)$ . We then use the first N features with the most frequent changes from the resulting feature frequency list.

Veldall (2008) also experiments with this selection method, and suggests to apply frequency-based selection to fluency ranking models that will be distributed to the public (for compactness' sake). In the variant he and Malouf and Van Noord (2004) discuss, all features that change within more than  $n$  contexts are included in the model.

## 2.4 Correlation-based Selection

While frequency-based selection helps selecting features that are discriminative, it cannot account for feature overlap. Discriminative features that have a strong correlation to features that were selected previously may still be added.

To detect overlap, we calculate the correlation of a candidate feature and exclude the feature if it shows a high correlation with features selected previously. To estimate Pearson's correlation of two features, we calculate the sample correlation coefficient,

$$r_{f_1, f_2} = \frac{\sum_{x \in X, y \in Y} (f_1(x, y) - \bar{f}_1)(f_2(x, y) - \bar{f}_2)}{(n-1)s_{f_1}s_{f_2}} \quad (2)$$

where  $\bar{f}_x$  is the average feature value of  $f_x$ , and  $s_{f_x}$  is the sample standard deviation of  $f_x$ .

Of course, correlation can only indicate overlap, and is in itself not enough to find effective features. In our experiments with correlation-based selection we used frequency-based selection as described in 2.3, to make an initial ranking of feature effectiveness.

## 2.5 Maximum Entropy Feature Selection

Correlation-based selection can detect overlap, however, there is yet another spurious type of feature that may reduce its effectiveness. Features with relatively noisy values may contribute less than their frequency of change may seem to indicate. For instance, consider a feature that returns a completely random value for every context. Not only does this feature change very often, its correlation with other features will also be weak. Such a feature may seem attractive from the point of view of a frequency or correlation-based method, but is useless in practice.

To account for both problems, we have to measure the effectiveness of features in terms of how much their addition to the model can improve prediction

of the training sample. Or in other words: does the log-likelihood of the training data increase?

We have modified the Selective Gain Computation (SGC) algorithm described by Zhou et al. (2003) for ranking tasks rather than classification tasks. This method builds upon the maximum entropy feature selection method described by Berger et al. (1996). In this method features are added iteratively to a model that is initially uniform. During each step, the feature that provides the highest gain as a result of being added to the model, is selected and added to the model.

In maximum entropy modeling, the weights of the features in a model are optimized simultaneously. However, optimizing the weights of the features in model  $p_{S, f}$  for every candidate feature  $f$  is computationally intractable. As a simplification, it is assumed that the weights of features that are already in the model are not affected by the addition of a feature  $f$ . As a result, the optimal weight  $\alpha$  of  $f$  can be found using a simple line search method.

However, as Zhou et al. (2003) note, there is still an inefficiency in that the weight of every candidate feature is recalculated during every selection step. They observe that gains of remaining candidate features rarely increase as the result of adding a feature. If it is assumed that this never happens, a list of candidate features ordered by gain can be kept. To account for the fact that the topmost feature in that list may have lost its effectiveness as the result of a previous addition of a feature to the model, the gain of the topmost feature is recalculated and reinserted into the list according to its new gain. When the topmost feature retains its position, it is selected and added to the model.

Since we use feature selection with features that are not binary, and for a ranking task, we modified the recursive forms of the model to:

$$\begin{aligned} \text{sum}_{S \cup f}^\alpha(y|x) &= \text{sum}_S(y|x) \cdot e^\alpha f(y) & (3) \\ Z_{S \cup f}^\alpha(x) &= Z_S(x) - \sum_y \text{sum}_S(y|x) \\ &\quad + \sum_y \text{sum}_{S \cup f}(y|x) & (4) \end{aligned}$$

Another issue that needs to be dealt with is the calculation of context and event probabilities. In the

literature two approaches are prevalent. The first approach divides the probability mass uniformly over contexts, and the probability of events within a context is proportional to the event score (Osborne, 2000):

$$p(x) = \frac{1}{|X|} \quad (5)$$

$$p(y|x) = \frac{p(x)}{\left(\frac{\text{score}(x,y)}{\sum_y \text{score}(x,y)}\right)} \quad (6)$$

where  $|X|$  is the number of contexts. The second approach puts more emphasis on the contexts that contain relatively many events with high scores, by making the context probability dependent on the scores of events within that context (Malouf and van Noord, 2004):

$$p(x) = \frac{\sum_y \text{score}(x,y)}{\sum_{y \in X} \text{score}(x,y)} \quad (7)$$

In our experiments, the second definition of context probability outperformed the first by such a wide margin, that we only used the second definition in the experiments described in this paper.

## 2.6 A Note on Overlap Detection

Although maximum-entropy based feature-selection may be worthwhile in itself, the technique can also be used during feature engineering to find overlapping features. In the selection method of Berger et al. (1996), the weight and gain of each candidate feature is re-estimated during each selection step. We can exploit the changes in gains to detect overlap between a selected feature  $f_n$ , and the candidates for  $f_{n+1}$ . If the gain of a feature changed drastically in the selection of  $f_{n+1}$  compared to that of  $f_n$ , this feature has overlap with  $f_n$ .

To determine which features had a drastic change in gain, we determine whether the change has a significance with a confidence interval of 99% after normalization. The normalized gain change is calculated in the following manner as described in algorithm 1.

---

**Algorithm 1** Calculation of the normalized gain delta

---

```

 $\Delta G_f \leftarrow G_{f,n} - G_{f,n-1}$ 
if  $\Delta G_f \geq 0.0$  then
   $\Delta G_{f,norm} \leftarrow \frac{\Delta G_f}{G_{f,n}}$ 
else
   $\Delta G_{f,norm} \leftarrow \frac{\Delta G_f}{G_{f,n-1}}$ 
end if

```

---

## 3 Experimental Setup

### 3.1 Task

We evaluated the feature selection methods in conjunction with a sentence realizer for Dutch. Sentences are realized with a chart generator for the Alpino wide-coverage grammar and lexicon (Bouma et al., 2001). As the input of the chart generator, we use abstract dependency structures, which are dependency structures leaving out information such as word order. During generation, we store the compressed derivation trees and associated (HPSG-inspired) attribute-value structures for every realization of an abstract dependency structure. We then use feature templates to extract features from the derivation trees. Two classes of features (and templates) can be distinguished *output features* that model the output of a process and *construction features* that model the process that constructs the output.

#### 3.1.1 Output Features

Currently, there are two output features, both representing auxiliary distributions (Johnson and Riezler, 2000): a word trigram model and a part-of-speech trigram model. The part-of-speech tag set consists of the Alpino part of speech tags. Both models are trained on newspaper articles, consisting of 110 million words, from the Twente Nieuws Corpus<sup>1</sup>.

The probability of unknown trigrams is estimated using linear interpolation smoothing (Brants, 2000). Unknown word probabilities are determined with Laplacian smoothing.

---

<sup>1</sup><http://wwwhome.cs.utwente.nl/druid/TwNC/TwNC-main.html>

### 3.1.2 Construction Features

The construction feature templates consist of templates that are used for parse disambiguation, and templates that are specifically targeted at generation. The parse disambiguation features are used in the Alpino parser for Dutch, and model various linguistic phenomena that can indicate preferred readings. The following aspects of a realization are described by parse disambiguation features:

- Topicalization of (non-)NPs and subjects.
- Use of long-distance/local dependencies.
- Orderings in the middle field.
- Identifiers of grammar rules used to build the derivation tree.
- Parent-daughter combinations.

Output features for parse disambiguation, such as features describing dependency triples, were not used. Additionally, we use most of the templates described by Velldal (2008):

- Local derivation subtrees with optional grand-parenting, with a maximum of three parents.
- Local derivation subtrees with back-off and optional grand-parenting, with a maximum of three parents.
- Binned word domination frequencies of the daughters of a node.
- Binned standard deviation of word domination of node daughters.

### 3.2 Data

The training and evaluation data was constructed by parsing 11764 sentences of 5-25 tokens, that were randomly selected from the (unannotated) Dutch Wikipedia of August 2008<sup>2</sup>, with the wide-coverage Alpino parser. For every sentence, the best parse according to the disambiguation component was extracted and considered to be correct. The Alpino system achieves a concept accuracy of around 90% on common Dutch corpora (Van Noord, 2007). The

<sup>2</sup><http://ilps.science.uva.nl/WikiXML/>

original sentence is considered to be the best realization of the abstract dependency structure of the best parse.

We then used the Alpino chart generator to construct derivation trees that realize the abstract dependency structure of the best parse. The resulting derivation trees, including attribute-value structures associated with each node, are compressed and stored in a derivation treebank. Training and testing data was then obtained by extracting features from derivation trees stored in the derivation treebank. At this time, the realizations are also scored using the General Text Matcher method (GTM) (Melamed et al., 2003), by comparing them to the original sentence. We have previously experimented with ROUGE-N scores, which gave rise to similar results. However, it is shown that GTM shows the highest correlation with human judgments (Cahill, 2009).

### 3.3 Methodology

To evaluate the feature selection methods, we first train models for each selection method in three steps: 1. For each abstract dependency structure in the training data 100 realizations (and corresponding features) are randomly selected. 2. Feature selection is applied, and the  $N$ -best features according to the selection method are extracted. 3. A maximum entropy model is trained using the TADM<sup>3</sup> software, with a  $\ell_2$  prior of 0.001, and using the  $N$ -best features.

We used 5884 training instances (abstract dependency trees, and scored realizations) to train the model. The maximum entropy selection method was used with a weight convergence threshold of  $1e^{-6}$ . Correlation is considered to be strong enough for overlap in the correlation-based method when two features have a correlation coefficient of  $r_{f_1, f_2} \geq 0.9$ .

Each model is then evaluated using 5880 held-out evaluation instances, where we select only instances with 5 or more realizations (4184 instances), to avoid trivial ranking cases. For every instance, we select the realization that is the closest to the original sentence to be the correct realization<sup>4</sup>. We then calculate the fraction of instances for which the

<sup>3</sup><http://tadm.sourceforge.net/>

<sup>4</sup>We follow this approach, because the original sentence is not always exactly reproduced by the generator.

model picked the correct sentence. Of course, this is a fairly strict evaluation, since there may be multiple equally fluent sentences.

## 4 Results

### 4.1 Comparing the Candidates

Since each feature selection method that we evaluated gives us a ranked list of features, we can train models for an increasing number of features. We have followed this approach, and created models for each method, using 100 to 5000 features with a step size of 100 features. Figure 2 shows the accuracy for all selection methods after  $N$  features. We have also added the line that indicates the accuracy that is obtained when a model is trained with all features (490667 features).

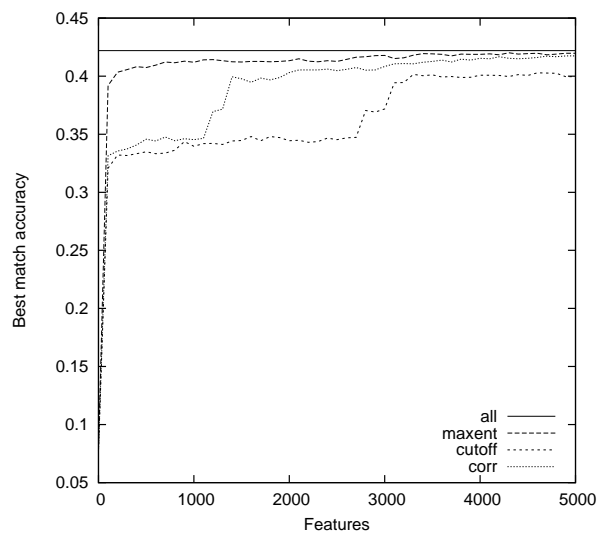


Figure 2: Accuracy of maximum entropy, correlation-based, and frequency-based selection methods after selecting  $N$  features ( $N \leq 5000$ ), with increments of 100 features.

In this graph we can see two interesting phenomena. First of all, only a very small number of features is required to perform this task almost as well as a model with all extracted features. Secondly, the maximum entropy feature selection model is able to select the most effective features quickly - fewer than 1000 features are necessary to achieve a relatively high accuracy.

As expected, the frequency-based method fared worse than maximum entropy selection. Initially

some very useful features, such as the  $n$ -gram models are selected, but improvement of accuracy quickly stagnates. We expect this to be caused by overlap of newly selected features with features that were initially selected. Even after selecting 5000 features, this method does not reach the same accuracy as the maximum entropy selection method had after selecting only a few hundred features.

The correlation-based selection method fares better than the frequency-based method without overlap detection. This clearly shows that feature overlap is a problem. However, the correlation-based method does not achieve good accuracy as quickly as the maximum entropy selection method. There are three possible explanations for this. First, there may be noisy features that are frequent, and since they show no overlap with selected features they are good candidates according to the correlation-based method. Second, less frequent features that overlap with a frequent feature in a subset of contexts may show a low correlation. Third, some less frequent features may still be very discriminative for the contexts where they appear, while more frequent features may just be a small indicator for a sentence to be fluent or non-fluent. It is possible to refine the correlation-based method to deal with the second class of problems. However, the lack of performance of the correlation-based method makes this unattractive - during every selection step a candidate feature needs to be compared with all previously selected features, rather than some abstraction of them.

Table 1 shows the peak accuracies when selecting up to 5000 features with the feature selection methods described. Accuracy scores of the random selection baseline, the  $n$ -gram models, and a model trained on all features are included for comparison. The random selection baseline picks a realization randomly. The  $n$ -gram models are the very same  $n$ -gram models that were used as auxiliary distributions in the feature-based models. The combined word/tag  $n$ -gram model was created by training a model with both  $n$ -gram models as the only features. We also list a variation of the frequency-based method often used in other work (such as Velldal (2008) and Malouf and Van Noord (2004)), where there is a fixed frequency threshold (here 4), rather than using the first  $N$  most frequently changing features.



Besides confirming the observation that feature selection can compress models very well, this table shows that the popular method of using a frequency cutoff, still gives a lot of opportunity for compressing the model further. In practice, it seems best to plot a graph as shown in figure 2, choose an acceptable accuracy, and to use the (number of) features that can provide that accuracy.

Method	Features	Accuracy
Random	0	0.0778
Tag n-gram	1	0.2039
Word n-gram	1	0.2799
Word/tag n-gram	2	0.2908
All	490667	0.4220
Fixed cutoff (4)	90103	0.4181
Frequency	4600	0.4029
Correlation	4700	0.4172
Maxent	4300	0.4201

Table 1: Peak accuracies for the maximum entropy, correlation-based, and frequency-based selection methods when selecting up to 5000 features. Accuracies for random, n-gram and full models are included for comparison.

## 4.2 Overlap in Frequency-based Selection

As we argued in section 2.5, the primary disadvantage of the frequency-based selection is that it cannot account for correlation between features. In the extreme case, we could have two very distinctive features  $f_1$  and  $f_2$  that behave exactly the same in any event. While adding  $f_2$  after adding  $f_1$  does not improve the model, frequency-based selection cannot detect this. To support this argumentation empirically, we analyzed the first 100 selected features to find good examples of this overlap.

Initially, the frequency-based selection chooses three distinctive features that are also selected by the maximum entropy selection method: the two n-gram language models, and a preference for topicalized NP subjects. After that, features that indicate whether the *vp\_arg\_v(np)* rule was used change very frequently within a context. However, this aspect of the parse tree is embodied in 13 successively selected features. Due to the generality of the feature templates, there are multiple templates to capture the use of this grammar rule: through local derivation

trees (with optional grandparenting), back-off for local derivation trees, and the features that calculate lexical node dominance.

Another example of such overlap in the first 100 features is in features modeling the use of the *non\_wh\_topicalization(np)* rule. Features containing this rule identifier are used 30 times in sequence, where it occurs in local derivation subtrees (with varying amounts of context), back-off local derivation subtrees, lexical node domination, or as a grandparent of another local derivation subtree.

In the first 100 features, there were many overlapping features, and we expect that this also is the case for more infrequent features.

## 4.3 Effective Features

The maximum entropy selection method shows that only a small number of features is necessary to perform fluency ranking (section 4.1). The first features that were selected in maximum entropy selection can give us good insight of what features are important for fluency ranking. Table 2 shows the 10 topmost features as returned by the maximum entropy selection. The weights shown in this table, are those given by the selection method, and their sign indicates whether the feature was characteristic of a fluent sentence (+) or a non-fluent sentence (-).

As expected (see table 1) the n-gram models are a very important predictor for fluency. The only surprise here may be that the overlap between both n-gram models is small enough to have both models as a prominent feature. While the tag n-gram model is a worse predictor than the word n-gram model, we expect that the tag n-gram model is especially useful for estimating fluency of phrases with word sequences that are unknown to the word n-gram model.

The next feature that was selected,  $r2(vp\_arg\_v(pred),2,vproj\_vc)$ , indicates that the rule *vp\_arg\_v(pred)* was used with a *vproj\_vc* node as its second daughter. This combination occurs when the predicative complement is placed after the copula, for instance as in *Amsterdam is de hoofdstad van Nederland* (*Amsterdam is the capital of The Netherlands*), rather than *De hoofdstad van Nederland is Amsterdam* (*The capital of The Netherlands is Amsterdam*).

The feature  $s1(non\_subj\_np\_topic)$  and its neg-

ative weight indicates that realizations with non-topicalized NP subjects are dispreferred. In Dutch, non-topicalized NP subjects arise in the OVS word-order, such as in *de soep eet Jan* (*the soup eats Jan*). While this is legal, SVO word-order is clearly preferred (*Jan eet de soep*).

The next selected feature (*ldsb(vc\_vb,vb\_v,[vproj\_vc,vp\_arg\_v(pp)])*) is also related to topicalization: it usually indicates a preference for prepositional complements that are not topicalized. For instance, *dit zorgde voor veel verdeeldheid* (*this caused lots of discord*) is preferred over the PP-topicalized *voor veel verdeeldheid zorgde dit* (*lots of discord caused this*).

*ldsb(n\_n\_pps,pp\_p\_arg(np),[])* gives preference PP-ordering in conjuncts where the PP modifier follows the head. For instance, the conjunct *groepen van bestaan of khandas* (*planes of existence or khandas*) is preferred by this feature over *van bestaan groepen of khandas* (*of existence planes or khandas*).

The next feature (*lds\_dl(mod2,[pp\_p\_arg(np)],[1],[non\_wh\_topicalization(modifier)])*) forms an exception to the dispreference of topicalization of PPs. If we have a PP that modifies a copula in a subject-predicate structure, topicalization of the PP can make the realization more fluent. For instance, *volgens Williamson is dit de synthese* (*according to Williamson is this the synthesis*) is considered more fluent than *dit is de synthese volgens Williamson* (*this is the synthesis according to Williamson*).

The final three features deal with punctuation. Since punctuation is very prevalent in Wikipedia texts due to the amount of definitions and clarifications, punctuation-related features are common. Note that the last two *lds\_dl* features may seem to be overlapping, they are not: they use different frequency bins for word domination.

## 5 Conclusions and Future Work

Our conclusion after performing experiments with feature selection is twofold. First, fluency models can be compressed enormously by applying feature selection, without losing much in terms of accuracy. Second, we only need a small number of targeted features to perform fluency ranking.

The maximum entropy feature selection method

Weight	Name
0.012	ngram_lm
0.009	ngram_tag
0.087	r2(vp_arg_v(pred),2,vproj_vc)
-0.094	s1(non_subj_np_topic)
0.090	ldsb(vc_vb,vb_v, [vproj_vc,vp_arg_v(pp)])
0.083	ldsb(n_n_pps,pp_p_arg(np),[])
0.067	lds_dl(mod2,[pp_p_arg(np)],[1], [non_wh_topicalization(modifier)])
0.251	lds_dl(start_start_ligg_streep, [top_start_xp,punct(ligg_streep), top_start_xp],[0,0,1],[top_start])
0.186	lds_dl(start_start_ligg_streep, [top_start_xp,punct(ligg_streep), top_start_xp],[0,0,2],[top_start])
0.132	r2(n_n_modroot(haak),5,1)

Table 2: The first 10 features returned by maximum entropy feature selection, including the weights estimated by this feature selection method.

shows a high accuracy after selecting just a few features. The commonly used frequency-based selection method fares far worse, and requires addition of many more features to achieve the same performance as the maximum entropy method. By experimenting with a correlation-based selection method that uses the frequency method to make an initial ordering of features, but skips features that show a high correlation with previously selected features, we have shown that the ineffectiveness of frequency-based selection can be attributed partly to feature overlap. However, the maximum entropy method was still more effective in our experiments.

In the future, we hope to evaluate the same techniques to parse disambiguation. We also plan to compare the feature selection methods described in this paper to selection by imposing a  $\ell_1$  prior.

The feature selection methods described in this paper are usable for feature sets devised for ranking and classification tasks, especially when huge sets of automatically extracted features are used. An open source implementation of the methods described in this paper is available<sup>5</sup>, and is optimized to work on large data and feature sets.

<sup>5</sup><http://daniel.dk.eu/Code/FeatureSqueeze/>

## References

- A.L. Berger, V.J.D. Pietra, and S.A.D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):71.
- G. Bouma, G. Van Noord, and R. Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In *Computational Linguistics in the Netherlands 2000. Selected Papers from the 11th CLIN Meeting*.
- T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.
- A. Cahill, M. Forst, and C. Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *ENLG '07: Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.
- A. Cahill. 2009. Correlating Human and Automatic Evaluation of a German Surface Realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 97–100.
- M. Johnson and S. Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 154–161, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- D. Madigan and A.E. Raftery. 1994. Model selection and accounting for model uncertainty in graphical models using Occam’s window. *Journal of the American Statistical Association*, 89(428):1535–1546.
- R. Malouf and G. van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *IJCNLP-04 Workshop: Beyond shallow analyses - Formalisms and statistical modeling for deep analyses*. JST CREST, March.
- I. D. Melamed, R. Green, and J. P. Turian. 2003. Precision and recall of machine translation. In *HLT-NAACL*.
- H. Nakanishi, Y. Miyao, and J. Tsujii. 2005. Probabilistic models for disambiguation of an hpsg-based chart generator. In *Parsing '05: Proceedings of the Ninth International Workshop on Parsing Technology*, pages 93–102, Morristown, NJ, USA. Association for Computational Linguistics.
- M. Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th conference on Computational linguistics*, pages 586–592, Morristown, NJ, USA. Association for Computational Linguistics.
- S. Perkins, K. Lacker, and J. Theiler. 2003. Grafting: Fast, incremental feature selection by gradient descent in function space. *The Journal of Machine Learning Research*, 3:1333–1356.
- A. Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1):151–175.
- G. Van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the 10th International Conference on Parsing Technologies*, pages 1–10. Association for Computational Linguistics.
- E. Velldal and S. Oepen. 2006. Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 517–525. Association for Computational Linguistics.
- E. Velldal. 2008. *Empirical Realization Ranking*. Ph.D. thesis, University of Oslo, Department of Informatics.
- Y. Zhou, F. Weng, L. Wu, and H. Schmidt. 2003. A fast algorithm for feature selection in conditional maximum entropy modeling.



# Short Papers



# Extracting Parallel Fragments from Comparable Corpora for Data-to-text Generation

Anja Belz                  Eric Kow

Natural Language Technology Group  
 School of Computing, Mathematical and Information Sciences  
 University of Brighton  
 Brighton BN2 4GJ, UK  
 {asb, eykk10}@bton.ac.uk

## Abstract

Building NLG systems, in particular statistical ones, requires parallel data (paired inputs and outputs) which do not generally occur naturally. In this paper, we investigate the idea of automatically extracting parallel resources for data-to-text generation from *comparable* corpora obtained from the Web. We describe our comparable corpus of data and texts relating to British hills and the techniques for extracting paired input/output fragments we have developed so far.

## 1 Introduction

Starting with Knight, Langkilde and Hatzivassiloglou's work on Nitrogen and its successor Halogen (Knight and Hatzivassiloglou, 1995; Knight and Langkilde, 2000), NLG has over the past 15 years moved towards using statistical techniques, in particular in surface realisation (Langkilde, 2002; White, 2004), referring expression generation (most of the systems submitted to the TUNA and GREC shared task evaluation challenges are statistical, see Gatt et al. (2008), for example), and data-to-text generation (Belz, 2008).

The impetus for introducing statistical techniques in NLG can be said to have originally come from machine translation (MT),<sup>1</sup> but unlike MT, where parallel corpora of inputs (source language texts) and outputs (translated texts) occur naturally at least in some domains,<sup>2</sup> NLG on the whole has to use manually created input/output pairs.

Data-to-text generation (D2T) is the type of NLG that perhaps comes closest to having naturally occurring inputs and outputs at its disposal. Work in D2T has involved different domains including generating weather forecasts from meteorological

data (Sripada et al., 2003), nursing reports from intensive care data (Portet et al., 2009), and museum exhibit descriptions from database records (Isard et al., 2003; Stock et al., 2007); types of data include dynamic time-series data (e.g. medical data) and static database entries (museum exhibits).

While data and texts in the three example domains cited above do occur naturally, two factors mean they cannot be used directly as example corpora or training data for building D2T systems: one, most are not freely available to researchers (e.g. by simply being available on the Web), and two, more problematically, for the most part, there is no direct correspondence between inputs and outputs as there is, say, between a source language text and its translation. On the whole, naturally occurring resources of data and related texts are not strictly parallel, but are merely what has become known as *comparable* in the MT literature, with only a subset of data having corresponding text fragments, and other text fragments having no obvious corresponding data items. Moreover, data transformations may be necessary before corresponding text fragments can be identified.

In this report, we look at the possibility of automatically extracting parallel data-text fragments from comparable corpora in the case of D2T from static database records. Such a parallel data-text resource could then be used to train an existing D2T generation system, or even build a new statistical generator from scratch, e.g. using techniques from statistical MT (Belz and Kow, 2009). The steps involved in going from comparable data and text resources to generators that produce texts similar to those in the text resource are then as follows: (1) identify sources on the Web for comparable data and texts; (2) pair up data records and texts; (3) extract parallel fragments (sets of data fields paired with word strings); (4) train a D2T generator using the parallel fragments; and (5) feed data inputs to the generator which then

<sup>1</sup>Nitrogen was conceived as an MT system component.

<sup>2</sup>Canadian and European parliamentary proceedings, etc.

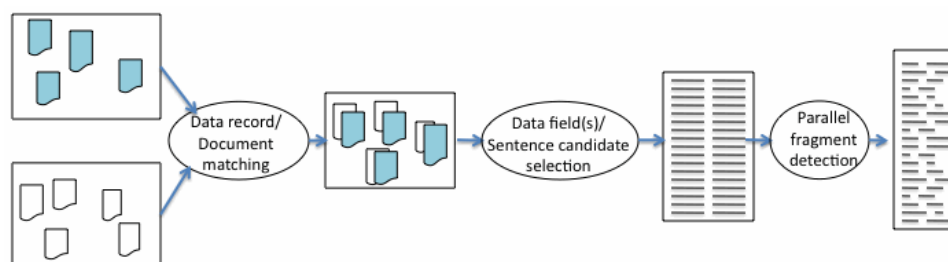


Figure 1: Overview of processing steps.

generates new texts describing them. Figure 1 illustrates steps 1–3 which this paper focuses on. In Section 3 we look at steps 1 and 2; in Section 4 at step 3. First we briefly survey related work in MT.

## 2 Related work in MT

In statistical MT, the expense of manually creating new parallel MT corpora, and the need for very large amounts of parallel training data, has led to a sizeable research effort to develop methods for automatically constructing parallel resources. This work typically starts by identifying comparable corpora. Much of it has focused on identifying word translations in comparable corpora, e.g. Rapp’s approach was based on the simple and elegant assumption that if words  $A_f$  and  $B_f$  have a higher than chance co-occurrence frequency in one language, then two appropriate translations  $A_e$  and  $B_e$  in another language will also have a higher than chance co-occurrence frequency (Rapp, 1995; Rapp, 1999). At the other end of the spectrum, Resnik & Smith (2003) search the Web to detect web pages that are translations of each other. Other approaches aim to identify pairs of sentences (Munteanu and Marcu, 2005) or sub-sentential fragments (Munteanu and Marcu, 2006) that are parallel within comparable corpora.

The latter approach is particularly relevant to our work. They start by translating each document in the source language (SL) word for word into the target language (TL). The result is given to an information retrieval (IR) system as a query, and the top 20 results are retained and paired with the given SL document. They then obtain all sentence pairs from each pair of SL and TL documents, and discard those sentence pairs with few words that are translations of each other. To the remaining sentences they then apply a fragment detection method which tries to distinguish between source fragments that have a translation on the target side, and fragments that do not.

The biggest difference between the MT situation and the D2T situation is that in the latter sentence-aligned parallel resources exist and can be used as a starting point. E.g. Munteanu & Marcu use an existing parallel Romanian-English corpus to (automatically) create a lexicon from which is then used in various ways in their method.

In D2T we have no analogous resources to help us get started, and the methods described in this paper use no such prior knowledge.

## 3 A Comparable Corpus of British Hills

As a source of data, we use the Database of British Hills (BHDB) created by Chris Crocker,<sup>3</sup> version 11.3, which currently contains measurements and other information about 5,614 British hills. Additionally, we perform reverse geocoding via the Google Map API<sup>4</sup> which allows us to convert latitude and longitude information from the hills database into country and region names. We add the latter to each database entry.

On the text side, we use Wikipedia texts in the WikiProject British and Irish Hills (retrieved on 2009-11-09). There are currently 899 pages covered by this WikiProject, 242 of which are of quality category B or above.<sup>5</sup>

**Matching up data records and documents:** Matching up the data records in the BHDB with articles in Wikipedia is not trivial: not all BHDB entries have corresponding Wikipedia articles, different hills often share the same name, and the same hill can have different names and spellings.

We perform a search of Wikipedia with the hill’s name as the search term, using the Mediawiki API, and then retain the top  $n$  search results returned (currently  $n = 1$ ). The top search result is not always a correct match for the database record. We

<sup>3</sup><http://www.biber.fsnet.co.uk>

<sup>4</sup><http://code.google.com/apis/maps/>

<sup>5</sup>B = The article is mostly complete and without major issues, but requires some further work.



```
{ "id": 1679, "main-name-info": { "name": "Hill of Stake", "notes": "",
                                "parent": "", "parent-notes": "" },
  "alt-name-info": [], "raw-name": "Hill of Stake", "rhb-section": "27A", "area": "Ayr to River Clyde",
  "height-metres": 522, "height-feet": 1713, "map-lto50k": "63", "map-lto25k": "341N", "gridref": "NS273630",
  "col-gridref": "NS320527", "col-height": 33, "drop": 489, "gridref10": "NS 27360 62998", "feature": "trig point",
  "observations": "", "survey": "", "date-climbed": "", "classification": "Ma,CoH,CoU",
  "county-name": "Renfrewshire(CoH); Renfrewshire(CoU)", "revision": "28-Oct-2001", "comments": "",
  "streetmap": "http://www.streetmap.co.uk/newmap.srf?x=227356&y=663005&z=3&sv=227356,663005&st=4&tl=~&bi=~&lu=N&ar=n",
  "ordnancesurvey-map": "http://getamap.ordnancesurvey.co.uk/getamap/frames.htm?mapAction=gaz&gazName=g&gazString=NS273630",
  "x-coord": 227356, "y-coord": 663005, "latitude": 55.82931,
  "longitude": -4.75789, "country": "Scotland", "region": "Renfrewshire" }
```

Hill of Stake is a hill on the boundary between North Ayrshire and Renfrewshire, Scotland. It is 522 metres (1712 feet) high. It is one of the Marilyns of Lowland Scotland. It is the highest point of the relatively low-lying county of Renfrewshire and indeed the entire Clyde Muirshiel Regional Park of which it is a part.

Table 1: Output of step 1: data record from British Hills DB and matched Wikipedia text (Hill of Stake).

manually selected the pairs we are confident are a correct match. This left us with 759 matched pairs out of a possible 899.

Table 1 shows an example of an automatically matched database entry and Wikipedia article. It illustrates the non-parallelism discussed in the preceding section; e.g. there is no information in the database corresponding to the last sentence.

## 4 Towards a Parallelised Corpus

### 4.1 Aligning data fields and sentences

In the second processing step, we pair up data fields and sentences. Related methods in MT have translation lexicons and thesauri that can be used as bridges between SL and TL texts, but there is no equivalents in NLG. Our current method associates each data field with a hand-written ‘match predicate’. For example, the match predicate for `height-metres` returns True if the sentence contains the words ‘X metres’ (among other patterns), where X is some number within 5% of the height of the hill in the database. We retain only the sentences that match at least one data field. Table 2 shows what the data field/sentence alignment procedure outputs for the Hill of Stake.

### 4.2 Identifying Parallel Fragments

While it was fine for step 2 to produce some rough matches, in step 3, parallel fragment detection, the aim is to retain only those parts of a sentence that can be said to realise some data field(s) in the set of data fields with which it has been matched.

**Computing data-text associations:** Following some preprocessing of sentences where each occurrence of a hill’s name and height is replaced by lexical class tokens `_NAME_`, `_HEIGHT_METRES_` or `_HEIGHT_FEET_`, the first step is to construct a

kind of lexicon of pairs  $(d, w)$  of data fields  $d$  and words  $w$ , such that  $w$  is often seen in the realisation of  $d$ . For this purpose we adapt Munteanu & Marcu’s (2006) method for (language to language) lexicon construction. For this purpose we compute a measure of the strength of association between data fields and words; we use the  $G^2$  log-likelihood ratio which has been widely used for this sort of purpose (especially lexical association) since it was introduced to NLP (Dunning, 1993). Following Moore (2004a) rather than Munteanu & Marcu, our current notion of cooccurrence is that a data field and word cooccur if they are present in the same pair of data fields and sentence (as identified by the method described in Section 4.1 above). We then obtain counts for the number of times each word cooccurs with each data field, and the number of times it occurs without the data field being present (and conversely). This allows us to compute the  $G^2$  score, for which we use the formulation from Moore (2004b) shown in Figure 2.

If the  $G^2$  score for a given  $(d, w)$  pair is greater than  $p(d)p(w)$ , then the association is taken to be positive, i.e.  $w$  is likely to be a realisation of  $d$ , otherwise the association is taken to be negative, i.e.  $w$  is likely not to be part of a realisation of  $d$ .

For each  $d$  we then convert  $G^2$  scores to probabilities by dividing  $G^2$  by the appropriate normalising factor (the sum over all negative  $G^2$  scores for  $d$  for obtaining the negative association probabilities, and analogously for positive associations). Table 3 shows the three words with the highest positive association probabilities for each of our six data fields. Note that these are not the three most likely alternative ‘translations’ of each data key, but rather the three words which are most likely to be part of a realisation of a data field, if seen in conjunction with it.

"main-name-only": "Hill of Stake", "country": "Scotland"	_NAME_ is a hill on the boundary between North Ayrshire and Renfrewshire, Scotland.
"height-metres": 522, "height-feet": 1713	It is _HEIGHT_METERS_ metres (_HEIGHT_FEET_ feet) high.
"country": "Scotland", "classification": ["Ma", "CoH", "CoU"]	It is one of the Marilyn's of Lowland Scotland.
"main-name-only": "Hill of Stake"	It is the highest point of the relatively low-lying county of Renfrewshire and indeed the entire Clyde Muirshiel Regional Park of which it is a part.

Table 2: Output of step 2: aligned data fields and sentences, for Hill of Stake.

$$2N \left( p(d, w) \log \frac{p(d, w)}{p(d)p(w)} + p(d, \neg w) \log \frac{p(d, \neg w)}{p(d)p(\neg w)} + p(\neg d, w) \log \frac{p(\neg d, w)}{p(\neg d)p(w)} + p(\neg d, \neg w) \log \frac{p(\neg d, \neg w)}{p(\neg d)p(\neg w)} \right)$$

Figure 2: Formula for computing  $G^2$  from Moore (2004b) ( $N$  is the sample size).

Data key $d$	Word $w$	$P^+(w d)$
main-name-only	_NAME_	0.1355
	a	0.0742
	in	0.0660
classification	as	0.0412
	adjoining	0.0193
	qualifies	0.0177
region	District	0.1855
	Lake	0.1661
	area	0.1095
country	in	0.1640
	_NAME_	0.1122
	Scotland	0.0732
height-metres	metres	0.1255
	m	0.0791
	height	0.0679
height-feet	feet	0.1511
	_HEIGHT_METERS_	0.0974
	(	0.0900

Table 3: Data keys with 3 most likely words.

**Identifying realisations:** The next step is to apply these probabilities to identify those parts of a sentence that are likely to be a valid realisation of the data fields in the input. In Figure 3 we plot the positive and negative association probabilities for one of the sentences from our running example, Hill of Stake. The light grey graph represents the association probabilities between each word in the sentence and `height-feet`, the dark grey line those between the words in the sentence and `height-metres`. We plot the negative association probabilities simply by multiplying each by  $-1$ .

The part of the sentence that one would want to extract as a possible realisation of  $\{\text{height-metres}, \text{height-feet}\}$ , namely “\_HEIGHT\_METRES\_ metres ( \_HEIGHT\_FEET\_ feet ) high”, shows up clearly as a sequence of relatively strong positive association values. Our current approach identifies such contiguous positive

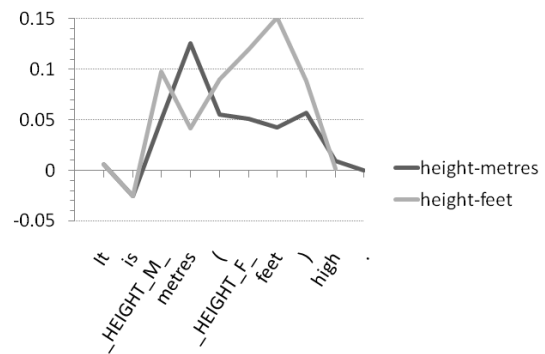


Figure 3: Positive and negative association probabilities plotted against the words they were computed for.

association scores and extracts the corresponding sentence fragments. This works well in many cases, but is too simple as a general approach; we are currently developing this method further.

## 5 Concluding Remarks

In this paper we have been interested in the problem of automatically obtaining parallel corpora for data-to-text generation. We presented our comparable corpus of 759 paired database entries and human-authored articles about British Hills. We described the three techniques which we have implemented so far and which we combine to extract parallel data-text fragments from the corpus: (i) identification of candidate pairs of data fields and sentences; (ii) computing scores for the strength of association between data and words; and (iii) identifying sequences of words in sentences that have positive association scores with the given data fields.

## References

- Anja Belz and Eric Kow. 2009. System building cost vs. output quality in data-to-text generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 16–24.
- A. Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.
- Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 1:61–74.
- A. Gatt, A. Belz, and E. Kow. 2008. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the 5th International Natural Language Generation Conference (INLG'08)*, pages 198–206.
- A. Isard, J. Oberlander, I. Androutsopoulos, and C. Matheson. 2003. Speaking the users' languages. 18(1):40–45.
- K. Knight and V. Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL '95)*.
- Kevin Knight and Irene Langkilde. 2000. Preserving ambiguity in generation via automata intersection. In *Proceedings of AAAI/IAAI*, pages 697–702.
- I. Langkilde. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. 2nd International Natural Language Generation Conference (INLG '02)*.
- Robert C. Moore. 2004a. Improving ibm word-alignment model 1. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 519–526.
- Robert C. Moore. 2004b. On log-likelihood-ratios and the significance of rare events. In *Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, pages 333–340.
- Dragos Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504.
- Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (COLING-ACL'06)*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.
- F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173:789–816.
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 320–322, Morristown, NJ, USA. Association for Computational Linguistics.
- Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 519–526, Morristown, NJ, USA. Association for Computational Linguistics.
- Philip Resnik and Noah Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29:349–380.
- S. Sripada, E. Reiter, J. Hunter, and J. Yu. 2003. Exploiting a parallel text-data corpus. In *Proceedings of Corpus Linguistics 2003*, pages 734–743.
- Oliviero Stock, Massimo Zancanaro, Paolo Busetta and Charles Callaway, Anbtonio Krüger, Michael Kruppa, Tsvi Kuflik, Elena Not, and Cesare Rocchi. 2007. Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction*, 17(3):257–304.
- M. White. 2004. Reining in CCG chart realization. In A. Belz, R. Evans, and P. Piwek, editors, *Proceedings INLG'04*, volume 3123 of *LNAI*, pages 182–191. Springer.



# Generating Natural Language Descriptions of Z Test Cases

**Maximiliano Cristiá**

Flowgate Consulting and CIFASIS  
Rosario, Argentina  
mcristia@flowgate.net

**Brian Plüss**

Centre for Research in Computing  
The Open University  
Milton Keynes, UK  
b.pluss@open.ac.uk

## Abstract

Critical software most often requires an independent validation and verification (IVV). IVV is usually performed by domain experts, who are not familiar with specific, many times formal, development technologies. In addition, model-based testing (MBT) is a promising testing technique for the verification of critical software. Test cases generated by MBT tools are logical descriptions. The problem is, then, to provide natural language (NL) descriptions of these test cases, making them accessible to domain experts. In this paper, we present ongoing research aimed at finding a suitable method for generating NL descriptions from test cases in a formal specification language. A first prototype has been developed and applied to a real-world project in the aerospace sector.

## 1 Introduction

Model-based testing (MBT) is an active research area and a promising theory of software and hardware testing (Utting and Legeard, 2006; Hierons et al., 2009). MBT approaches start with a formal model or specification of the software, from which test cases are generated. These techniques have been developed and applied to models written in different formal notations, such as Z (Stocks and Carrington, 1996), finite state machines and their extensions (Grieskamp et al., 2002), B (Legeard et al., 2002), algebraic specifications (Bernot et al., 1991), and so on.

The fundamental hypothesis behind MBT is that, as a program is correct if it verifies its specification, then the specification is an excellent source of test cases. Once test cases are derived from the model, they are refined to the level of the implementation language and executed. The resulting

output is then abstracted to the level of the specification language, and the model is used again to verify if the test case has detected an error.

The Test Template Framework (TTF) described by Stocks and Carrington (1996) is a particular MBT theory specially well suited for unit testing. The TTF uses Z specifications (Spivey, 1989) as the entry models and prescribes how to generate test cases for each operation included in the model. Fastest (Cristiá and Rodríguez Monetti, 2009) implements the TTF allowing users to automatically produce test cases for a given Z specification. Recently, we used Fastest to test an on-board satellite software for a major aerospace company in South America. Since Fastest uses models written in the Z specification language, test cases generated by this tool are paragraphs of formal text (see Section 2). This description is suitable for the automatic tasks involved in testing (e.g., automatic execution, hyperlinking, traceability), but humans need to be able to read Z specifications in order to understand what is being tested. In projects where independent verification and validation (IVV) is required this might be a problem, as most stakeholders will not necessarily be fluent in Z.

This is precisely the case in the project mentioned above, where the aerospace company requested not only the test cases in Z, but also in English. As it can be expected, in a project with hundreds of test cases, manual translation would increase the overall cost of testing and, most critically, reduce its quality due to the introduction of human errors. Interestingly, this problem is opposite to those in mainstream industrial practice, where test cases are described in natural language and must be formalised, in order to augment the quality and, hopefully, reduce the costs of testing.

Given the formal, structured nature of the source text, natural language generation (NLG) techniques seem to be an appropriate approach to solving this problem. In the rest of the pa-

per, we give an example of a test case from the project mentioned above (Section 2), describe a template-based method for generating NL descriptions (Section 3), and propose further work towards a more general NLG solution (Section 4).

## 2 An Example from the Aerospace Industry

The problem of generating NL descriptions of specifications in Z arises in the following scenario: a company developing the software for a satellite needs to verify that the implementation conforms to a certain aerospace standard (ECSS, 2003) describing the basic functionality of any satellite software. We therefore started by modelling in Z the services described by the standard and used the Fastest tool to generate test cases.

The model is a “standard” Z specification: it has a *schema box* that defines the state space of the system and *operations* defining the transition relation between states<sup>1</sup>. Each operation formalizes one of the services described by the standard (e.g., memory dump, telecommand verification, enabling or disabling on-board monitoring, etc.).

Figure 1 shows one of the test cases generated for the operation *DumpMemoryAbsAdd*, that models a remote request for the on-board software to dump some portion of its memory. In TTF and Fastest, a Z test case is essentially a set of bindings between variables and values, and test cases are grouped according to the operation they test. Identifiers appearing in a test case are the input and state variables from the definition of the operation. These are bound to certain values defining the state in which the system must be tested and the input given to the operation in each unique test case. In the example, input variables are those *decorated* with a question mark, while state variables are plain identifiers. All these variables are declared somewhere else in the specification, by using a special schema box called *valid input space*, associated with each operation.

For example, the Z schema in Figure 1 indicates that the implementation of the dump memory service must be tested in a state where the system is processing a telecommand (*processingTC = yes*), the telecommand is a request for a memory dump (*srv = DMMA*), the system has one memory block

```

DumpMemoryAbsAdd_SP_7_TCASE
mid = mid0 ∧ srv = DMMA ∧ lengths = ∅
processingTC = yes ∧ adds = ∅
blocks = {mid0 ↦ {1 ↦ byte0, 2 ↦ byte1,
                 3 ↦ byte2, 4 ↦ byte3}}
m? = mid0 ∧ sa? = ⟨1⟩ ∧ len? = ⟨2⟩

```

Figure 1: A test case described in Z

which is four bytes long (*blocks = {...}*), there are no other pending requests (*adds = lengths = ∅*); and the request is for a memory dump of length two (*len? = ⟨2⟩*) starting at the first address (*sa? = ⟨1⟩*) of the available memory block (*m? = mid0 = mid*).

Fastest generated almost 200 test cases like the one depicted in Figure 1 from a model describing a simplified version of five services listed in the standard. The customer requested to deliver a natural language description of each one of them and a model describing all the services would have thousands of test cases. Clearly, trying to make the translation by hand would have been not only a source of errors, but also a technical retreat.

## 3 A Template-Based NLG Solution

As a first approach, we used a template-based method. We started by defining a grammar to express what we called *NL test case templates* (NLTCT). It appears in Figure 2<sup>2</sup>. Each NLTCT specifies how an NL description is generated for the Z test cases of a given operation. It starts with the name of the operation. Next follows a text section, intended as a parametrized NL description of the test case, where calls to *translation rules* can be inserted as needed. Finally, all necessary translation rules are defined, by indicating what is written in replacement for a call when a certain variable in the formal description of a test case appears bound to a specific value. In this way, a different text is generated for each test case, according to the binding between values and variables that defines the case. The Appendix shows the NLTCT for the operation *DumpMemoryAbsAdd*.

We implemented a parser in *awk* that takes an NLTCT and a Z test case, and generates the NL description of each test case in the input. Figure 3 shows the result for the test case in Figure 1.

This first prototype showed that NLTCTs tend

<sup>1</sup>The Z specification language is essentially typed first order logic, with syntactic sugar in the form of operators, that serve as shortcuts for frequently used complex expressions.

<sup>2</sup>Fastest saves formal test cases in text files written in ZLaTeX, an extension of the L<sup>A</sup>T<sub>E</sub>X markup language, what explains the use of this format in the NLTCT grammar.

```

NLCT ::= ⟨Operation⟩ eol
        ⟨NLCD⟩ eol
        ⟨TCRule⟩ {, ⟨TCRule⟩}
Operation ::= operation =⟨identifier⟩
NLCD ::= \begin{tcase} eol
        ⟨LaTeXText⟩ eol
        \end{tcase}
LaTeXText ::= LaTeX | ⟨TCRuleCall⟩ | ⟨LaTeXText⟩
TCRuleCall ::= & rule ⟨identifier⟩ &
TCRule ::= \begin{trule}{⟨identifier⟩} eol
        case ⟨identifier⟩[, ⟨identifier⟩] eol
        ⟨RuleDef⟩ eol {, ⟨RuleDef⟩ eol}
        endcase eol
        \end{trule}
RuleDef ::= $⟨ZLaTeX⟩[“ | “ ⟨ZLaTeX⟩ | & ⟨ZLaTeX⟩]
        : ⟨LaTeX⟩ eol
LaTeX ::= free LATEX text
ZLaTeX ::= free Z LATEX text

```

Figure 2: Grammar for NLCT templates

to be relatively small and simple, in spite of the large number of test cases. This is due to test cases combining a small set of values in many different ways. However, NLCTs for large operations tend to become increasingly more complex, for the number of combinations grows exponentially. As a consequence, these operations require a large number of cases within translation rules and sometimes even more translation rules<sup>3</sup>.

A thorough evaluation of this method is due. Its suitability must be measured from the perspective of two kinds of users: (a) the engineers who write the formal models, generate the formal test cases and write the NLCTs; and (b) other stakeholders (e.g., the customer, auditors, domain experts), who read the descriptions of the test cases in natural language. For the engineers, applying the method should be more efficient, in terms of time and effort, than writing the descriptions by hand. For the readers, success will be determined by the readability of the output and, more critically, by its precision with respect to the specification. At the moment of writing, we are designing two empirical studies aimed at obtaining these measures.

#### 4 Future and Related Work

The solution presented above was successful in generating adequate NL descriptions of the test

<sup>3</sup>This is because templates are written in terms of the values bound to variables, and not in terms of the predicates satisfied by those values, which are nonetheless available as part of the MBT approach.

#### Test case: DumpMemoryAbsAdd\_SP\_7\_TCASE

Service (6,5) will be tested in a situation that verifies that:

- the state is such that:
  - the on-board system is currently processing a telecommand and has not answered it yet.
  - the service type of the telecommand is DMAA.
  - the set of sequences of available memory cells contains only one sequence, associated to a memory ID, which has four different bytes.
  - the set of starting addresses of the chunks of memory that have been requested by the ground is empty.
- the input memory ID to be dumped is the available memory ID, the input set of start addresses of the memory regions to be dumped is the unitary sequence composed of 1, the set of numbers of memory cells to be dumped is the unitary sequence composed of 2.

Figure 3: NL description of the test in Figure 1

cases in one particular project. However, the limitations mentioned in the previous section show that this solution would not generalise well to specifications in other domains. Moreover, it requires defining a new template for each operation; a task of still considerable size for large systems.

At the same time, Z specifications contain all the information necessary to produce the templates for the operations in the system, regardless of its domain of application. This information is structured according to the syntax of the formal language. Additionally, when formally specifying a system, it is common practice to include associations between the identifiers in the specification (new types, operations, state schemata, variables, constants, etc.) and the elements they refer to in the application domain (i.e., aerospace software). These associations are called *designations* (Jackson, 1995), some of which, relevant to the test case in Figure 1, are shown in Figure 4.

These considerations lead us to believe in the

```

srv ≈ Service type of the telecommand
DMAA ≈ Dump memory using absolute addresses
processingTC ≈ The on-board system is currently processing
a telecommand and has not answered it yet
m? ≈ Memory ID to be dumped
sa? ≈ Start addresses of the memory regions to be
dumped
len? ≈ The number of memory cells to be dumped
for each start address

```

Figure 4: Designations for the test in Figure 1

possibility of generating NL descriptions of Z test cases automatically by using their definitions, the system specification and the designations of identifiers. Such a solution would be independent of the application domain and, more importantly, of the number of operations in the model.

The linguistic properties of the target document are relevant in devising an adequate treatment for the input, but the overall structure of the output remains rigid and its content is determined by the definition of each test case. The approach would still be template-based, but in terms of the NLG architecture of Reiter and Dale (2000), templates would be defined at the level of the document structure<sup>4</sup>, with minimal microplanning and surface strings generated according to the part of the test case being processed and the designations of the identifiers<sup>5</sup>. The next stages of our project will point in this direction, using techniques from NLG for automating the definition of the templates presented in the previous section.

There have been efforts for producing natural language versions of formal specifications in the past. Punshon et al. (1997) use a case study to present the REVIEW system (Salek et al., 1994)<sup>6</sup>. REVIEW automatically paraphrases specifications developed with Metaview (Sorenson et al., 1988), an academic research *metasystem* that facilitates the construction of CASE environments to support software specification tasks. Coscoy (1997) describes a mechanism based on program extraction, for generating explanations of formal proofs in the Calculus of Inductive Constructions, implemented in the Coq Proof Assistant (Bertot and Castéran, 2004). Lavoie et al. (1997) present MODEX, a tool that generates customizable descriptions of the relations between classes in object-oriented models specified in the ODL standard (Cattell and Barry, 1997). Bertani et al. (1999) describe a controlled natural language approach to translating formal specifications written in an extension of TRIO (Ghezzi et al., 1990) by transforming syntactic trees in TRIO into syntactic trees of the controlled language.

The solutions presented in the related work above are highly dependant on particular aspects

<sup>4</sup>Somewhat along the lines of what Wilcock (2005) describes for XML-based NLG.

<sup>5</sup>This approach is similar to the method proposed by Kittridge and Lavoie (1998) for generating weather forecasts.

<sup>6</sup>Salek et al. (1994) also give a comprehensive survey of related work for generating NL explanations for particular specification languages (most of which are now obsolete).

of the source language and do not apply directly to specifications written in Z. To our knowledge, no work has been done towards producing NL descriptions of Z specifications. The same holds for test cases generated using the MBT approach.

## 5 Conclusion

In this paper we presented a concrete NLG problem in the area of software development involving formal methods. We focused the description on the generation of NL descriptions of test cases, but nothing prevents us from extending the idea to entire system specifications.

The development of a general technique for verbalising formal specification would fill the communication gap between system designers and other stakeholders in the development process, while preserving the advantages associated to the use of formal methods: precision, lack of ambiguity, formal proof of system properties, etc.

Finally, we hope this paper draws attention from NLG experts to an area which would benefit substantially from their expertise.

## Acknowledgements

A substantial part of this research was funded by Flowgate Consulting. We would also like to thank Richard Power from the NLG group at The Open University for help in finding financial support, Eva Banik for helpful comments on earlier versions of this paper, and three anonymous reviewers for useful feedback and suggestions.

## References

- G. Bernot, M.C. Gaudel, and B. Marre. 1991. Software testing based on formal specifications: a theory and a tool. *Software Engineering Journal (SEJ)*, 6(6):387–405.
- A. Bertani, W. Castelnovo, E. Ciapessoni, and G. Mauri. 1999. Natural language translations of formal specifications for complex industrial systems. In *AI\*IA 1992: Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence*, pages 185–194, Bologna, Italy.
- Y. Bertot and P. Castéran. 2004. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. Springer-Verlag.
- R.G.G. Cattell and D.K. Barry, editors. 1997. *The object database standard: ODMG 2.0*. Morgan Kaufmann Publishers Inc., San Francisco, CA.



- Y. Coscoy. 1997. A natural language explanation for formal proofs. In *LACL '96: Selected papers from the First International Conference on Logical Aspects of Computational Linguistics*, pages 149–167, London, UK. Springer-Verlag.
- M. Cristiá and P. Rodríguez Monetti. 2009. Implementing and applying the Stocks-Carrington framework for model-based testing. In Karin Breitman and Ana Cavalcanti, editors, *ICFEM*, volume 5885 of *Lecture Notes in Computer Science*, pages 167–185. Springer-Verlag.
- ECSS. 2003. Space Engineering – Ground Systems and Operations: Telemetry and Telecommand Packet Utilization. Technical Report ECSS-E-70-41A, European Space Agency.
- C. Ghezzi, D. Mandrioli, and A. Morzenti. 1990. TRIO: A logic language for executable specifications of real-time systems. *Journal of Systems and Software*, 12(2):107–123.
- W. Grieskamp, Y. Gurevich, W. Schulte, and M. Veanes. 2002. Generating finite state machines from abstract state machines. In *ISSTA '02: Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 112–122, Rome, Italy.
- R.M. Hierons, K. Bogdanov, J.P. Bowen, R. Cleaveland, J. Derrick, et al. 2009. Using formal specifications to support testing. *ACM Computing Surveys (CSUR)*, 41(2):9.
- M. Jackson. 1995. *Software requirements & specifications: a lexicon of practice, principles, and prejudices*. Addison-Wesley.
- R. Kittredge and B. Lavoie. 1998. Meteocogent: A knowledge-based tool for generating weather forecast texts. In *Proceedings of American Meteorological Society AI Conference (AMS-98)*, Phoenix, AZ.
- B. Lavoie, O. Rambow, and E. Reiter. 1997. Customizable descriptions of object-oriented models. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, pages 253–256, Washington, DC.
- B. Legeard, F. Peureux, and M. Utting. 2002. A Comparison of the BTT and TTF Test-Generation Methods. In *ZB '02: Proceedings of the 2nd International Conference of B and Z Users on Formal Specification and Development in Z and B*, pages 309–329, London, UK. Springer-Verlag.
- J.M. Punshon, J.P. Tremblay, P.G. Sorenson, and P.S. Findeisen. 1997. From formal specifications to natural language: A case study. In *12th IEEE International Conference Automated Software Engineering*, pages 309–310.
- E. Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.
- A. Salek, P.G. Sorenson, J.P. Tremblay, and J.M. Punshon. 1994. The REVIEW system: From formal specifications to natural language. In *Proceedings of the First International Conference on Requirements Engineering*, pages 220–229.
- P.G. Sorenson, J.P. Tremblay, and A.J. McAllister. 1988. The Metaview system for many specification environments. *IEEE Software*, 5(2):30–38.
- J.M. Spivey. 1989. *The Z Notation: A Reference Manual*. Prentice-Hall, Inc.
- P. Stocks and D. Carrington. 1996. A Framework for Specification-Based Testing. *IEEE Transactions on Software Engineering*, 22(11):777–793.
- M. Utting and B. Legeard. 2006. *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA.
- G. Wilcock. 2005. An Overview of Shallow XML-based Natural Language Generation. In *Proceedings of the 2nd Baltic Conference on Human Language Technologies*, pages 67–78, Tallinn, Estonia.

## Appendix A. NLTCT for the Example

NLTCT for *DumpMemoryAbsAdd* (some parts were replaced by [...] due to space restrictions):

```
operation = DumpMemoryAbsAdd

\begin{tcase}
\centerline{\bf Test case: \ltcaseid}}

The service (6,5) will be tested in the
situation that verifies that:
\begin{itemize}
\item the state is such that:
\begin{itemize}
\item the on-board system is &trule PTCr&.
\item the service type of the telecommand
is &trule SRVr&.

[...]
\item the set of starting addresses of the
chunks of memory that have been
requested by the ground is &trule
ADSr&.
\end{itemize}
\end{itemize}
[...]
\end{itemize}
\end{tcase}

\begin{trule}{PTCr}
case processingTC
$yes :currently processing a telecommand and
has not answered it yet
$no :not currently processing a telecommand
endcase
\end{trule}

\begin{trule}{SRVr}
case srv
$* :*
endcase
\end{trule}

\begin{trule}{ADSr}
case adds
$\emptysetset :empty
$\langle 0 \rangle :the unitary sequence
composed of 0
endcase
\end{trule}

[...]
```



# Applying semantic frame theory to automate natural language template generation from ontology statements

Dana Dannélls

NLP research unit, Department of Swedish Language  
University of Gothenburg, SE-405 30 Gothenburg, Sweden  
dana.dannells@svenska.gu.se

## Abstract

Today there exist a growing number of framenet-like resources offering semantic and syntactic phrase specifications that can be exploited by natural language generation systems. In this paper we present on-going work that provides a starting point for exploiting framenet information for multilingual natural language generation. We describe the kind of information offered by modern computational lexical resources and discuss how template-based generation systems can benefit from them.

## 1 Introduction

Existing open-source multilingual natural language generators such as NaturalOWL (Galanis and Androutsopoulos, 2007) and MPIRO (Isard et al., 2003) require a large amount of manual linguistic input to map ontology statements onto semantic and syntactic structures, as exemplified in Table 1. In this table, each statement contains a property and two instances; each template contains the lexicalized, reflected property and the two ontology classes (capitalized) the statement's instances belong to.

Ontology statement	Sentence template
painted-by (ex14, p-Kleo)	VESSEL <i>was decorated by</i> PAINTER
exhibit-depicts (ex12, en914)	PORTRAIT <i>depicts</i> EXHIBIT-STORY
current-location (ex11, wag-mus)	COIN <i>is currently displayed in</i> MUSEUM

Table 1: MPIRO ontology statements and their corresponding sentence templates.

Consider adapting such systems to museum visitors in multilingual environments: as each statement is packaged into a sentence through a fixed sentence template, where lexical items, style of reference and linguistic morphology have already been determined, this adaptation process requires an extensive amount of manual input for each language, which is a labour-intensive task.

One way to automate this natural language mapping process, avoiding manual work is through language-specific resources that provide semantic and syntactic phrase specifications that are, for example, presented by means of lexicalized frames. An example of such a resource in which frame principles have been applied to the description and the analysis of lexical entries from a variety of semantic domains is the Berkeley FrameNet (FN) project (Fillmore et al., 2003). The outcome of the English FN has formed the basis for the development of more sophisticated and computationally oriented multilingual FrameNets that today are freely available (Boas, 2009).

This rapid development in computational lexicography circles has produced a growing number of framenet-like resources that we argue are relevant for natural language generators. We claim that semantic and syntactic information, such as that provided in a FrameNet, facilitates mapping of ontology statements to natural language. In this paper we describe the kind of information which is offered by modern computational lexical resources and discuss how template-based natural language generation (NLG) systems can benefit from them.

### 1.1 Semantic frames

A frame, according to Fillmore's frame semantics, describes the meaning of lexical units with reference to a structured background that motivates the conceptual roles they encode. Conceptual roles are represented with a set of slots called frame elements (FEs). A semantic frame carries information about the different syntactic realizations of the frame elements (syntactic valency), and about their semantic characteristics (semantic valency).

A frame can be described with the help of two types of frame elements that are classified in terms of how central they are to a particular frame, namely: core and peripheral. A core ele-

ment is one that instantiates a conceptually necessary component of a frame while making the frame unique and different from other frames. A peripheral element does not uniquely characterize a frame and can be instantiated in any semantically appropriate frame.

## 1.2 The language generation module

The kind of language generation system discussed here consists of a language generation module that is guided by linguistic principles to map its non-linguistic input (i.e. a set of logical statements) to syntactic and semantic templates. This kind of generation system follows the approaches that have been discussed elsewhere (Reiter, 1999; Busemann and Horacek, 1998; Geldof and van de Velde, 1997; Reiter and Mellish, 1993).

The goal of the proposed module is to associate an ontology statement with relevant syntactic and semantic specifications. This generation process should be carried out during microplanning (cf. Reiter and Dale (2000)) before aggregation and referring expression generation take place.

## 1.3 The knowledge representation

The knowledge representation which serves as the input to the language generator is a structured ontology specified in the Web Ontology Language (OWL) (Berners-Lee, 2004) on which programs can perform logical reasoning over data.

Ontological knowledge represented in OWL contains a hierarchical description of classes (concepts) and properties (relations) in a domain. It may also contain instances that are associated with particular classes, and assertions (axioms), which allow reasoning about them. Generating linguistic output from this originally non-linguistic input requires instantiations of the ontology content, i.e. concepts, properties and instances by lexical units.

## 2 From ontology statements to template specifications

Our approach to automatic template generation from ontology statements has three major steps: (1) determining the *base lexeme* of a statement's property and identifying the frame it evokes,<sup>1</sup> (2) matching the statement's associated concepts with the frame elements, and (3) extracting the syntactic patterns that are linked to each frame element.

<sup>1</sup>Base lexemes become words after they are subjected to morphological processing which is guided by the syntactic context.

The remainder of this section describes how base lexemes are chosen and how information about the syntactic and semantic distribution of the lexemes underlying an ontological statement are acquired.

## 2.1 Lexical units' determination and frame identification

The first, most essential step that is required for recognizing which semantic frame is associated with an ontology statement is lexicalization. Most Web ontologies contain a large amount of linguistic information that can be exploited to map the ontology content to linguistic units automatically (Mellish and Sun, 2006). However, direct verbalization of the ontology properties and concepts requires preprocessing, extensive linguistic knowledge and sophisticated disambiguation algorithms to produce accurate results. For the purposes of this paper where we are only interested in lexicalizing the ontology properties, we avoid applying automatic verbalization; instead we choose manual lexicalization.

The grammatical categories that are utilized to manifest the ontology properties are verb lexemes. These are determined according to the frame definitions and with the help of the ontology class hierarchy. For example, consider the statement *create (bellini, napoleon)*. In this domain, i.e. the cultural heritage domain, the property *create* has two possible interpretations: (1) to create a physical object which serves as the representation of the presented entity, (2) to create an artifact that is an iconic representation of an actual or imagined entity or event. FrameNet contains two frames that correspond to these two definitions, namely: *Create Representation* and *Create physical artwork*.

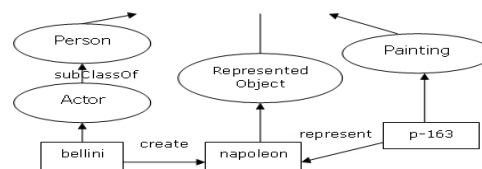


Figure 1: A fragment of the ontology.

By following the ontological representation departing from the given instances, as illustrated in Figure 1, we learn that *bellini* is an instance of the class *Actor*, *napoleon* is an instance of the class *Represented Object*, and that *napoleon* is the represented entity in the painting *p-163*. Thus, in this

context, an appropriate lexicalization of the property *create* is the verb *paint* which evokes the *Create Representation* frame.

For clarity, we specify in Table 2 part of the information that is coded in the frame. In this table we find the name of the frame, its definition, the set of lexical units belonging to the frame, the names of its core elements and a number of sentences annotated with these core FEs.

Create_representation	
Def	A Creator produces a physical object which is to serve as a Representation of an actual or imagined entity or event, the Represented.
LUs	carve.v, cast.v, draw.v, paint.v, photograph.v, sketch.v
core FEs	Creator (C)
	Represented (R)

Table 2: Frame *Create\_representation*.

## 2.2 Matching the ontology concepts with frame elements

In this step, the set of core frame elements which function as the obligatory arguments of the required lexeme are matched with their corresponding ontology concepts. The algorithm that is applied to carry out this process utilizes the FE Taxonomy and the ontology class hierarchy.<sup>2</sup>

Matching is based on the class hierarchies. For example: *Actor*, which is a subclass of *Person* is matched with the core element *Creator*, which is a subclass of *Agent* because they are both characterized as animate objects that have human properties. Similarly, *Represented\_Object*, which is a subclass of *Conceptual\_Object*, is matched with the core element *Represented*, which is a subclass of *Entity* because they are both characterized as the results of a human creation that comprises non-material products of the human mind.

This matching process leads to consistent specifications of the semantic roles specifying sentence constituents which are not bound to the input ontology structure.<sup>3</sup>

## 2.3 Semantic and syntactic knowledge extraction

Semantic frames, besides providing information about a lexeme's semantic content, provide information about the valency pattern associated with

<sup>2</sup>The Frame Element Taxonomy: <http://www.cires.com/db/feindex.html>

<sup>3</sup>One of the basic assumptions of our approach is that semantically, languages have a rather high degree of similarity, whereas syntactically they tend to differ.

it, i.e. how semantic roles are realized syntactically and what are the different types of grammatical functions they may fulfill when occurring with other elements. An example of the syntactic patterns and possible realizations of the semantic elements that appear in the *Create\_representation* frame are summarized in Table 3.<sup>4</sup> From this information we learn the kind of syntactic valency patterns that are associated with each semantic element. For example, we learn that in active constructions *Creator* appears in the subject position while in passive constructions it follows the preposition *by*. It can also be eliminated in passive constructions when other peripheral elements appear (Example 2), in this case it is the FE *Time* (T). Although it is a peripheral element, it plays an important role in this context.

FEs	Syntactic Pattern
[C, R]	[[NP <i>Ext</i> ], [NP <i>Obj</i> ]]
Example 1:	[Leonardo da Vinci] <sub>C</sub> painted [this scene] <sub>R</sub>
[R, T]	[[ [NP <i>Ext</i> ], PP[in] <i>Dep</i> ]]
Example 2:	[The lovely Sibyls] <sub>R</sub> were painted in [the last century] <sub>T</sub> .
[R, C, T]	[[ [NP <i>Ext</i> ], [PP[by] <i>Dep</i> ], [PP[in] <i>Dep</i> ]]
Example 3:	[The Gerichtsstube] <sub>R</sub> was painted by [Kuhn] <sub>C</sub> in [1763] <sub>T</sub> .

Table 3: Syntactic realizations of the lexical entry *paint*.

This knowledge is extracted automatically from the FN database and is converted to sentence specifications with the help of a simple Perl script. Below is a template example which specifies the sentence construction of the sentence in Example 3:

```
(template ( type: passive)
  (( head: |paint|) (feature: (tense: past) )
   ( arg1 (Represented (head: |gerichtsstube|) (
     determiner: |the|))
    arg2 (Creator (head: |kuhn|) (mod: |by|))
    arg3 (Time (head: |1763|) (mod: |in|))))
```

## 3 Testing the method

To test our approach, we employ the MPIRO domain ontology content.<sup>5</sup> Table 4 illustrates some of the results, i.e. examples of the ontology statements, the frame that matched their property lexicalization, and their possible realization patterns that were extracted from the English FrameNet.

The results demonstrate some of the advantages of the syntactic and semantic valency properties provided in FN that are relevant for expressing natural language. These include: Verb collocations

<sup>4</sup>FN's abbreviations: Constructional Null Instantiation (CNI), External Argument (Ext), Dependent (Dep).

<sup>5</sup><<http://users.iit.demokritos.gr/~eleon/ELEONDownloads.html>>

Nr	Ontology statement	Frame	Possible realization patterns
(1)	depict (portrait <sub>MED</sub> , story <sub>ITE</sub> )	Communicate_categorization	MEDIUM <i>depict</i> CATEGORY. MEDIUM <i>depict</i> ITEM of CATEGORY.
(2)	depict (modig <sub>CRE</sub> , portrait <sub>REP</sub> )	Create_physical_artwork	CREATOR <i>paint</i> REPRESENTATION. CREATOR <i>paint</i> REPRESENTATION <i>from</i> REFERENCE in PLACE.
(3)	depict (kuhn <sub>CRE</sub> , flower <sub>REP</sub> )	Create_representation	CREATOR <i>paint</i> REPRESENTED. REPRESENTED <i>is painted by</i> CREATOR in TIME.
(4)	locate (portrait <sub>THE</sub> , louvre <sub>LOC</sub> )	Being_located	THEME <i>is located</i> LOCATION.
(5)	copy (portrait <sub>ORI</sub> , portrait <sub>COP</sub> )	Duplication	COPY <i>replicate</i> ORIGINAL. CREATOR <i>replicate</i> ORIGINAL.

Table 4: Ontology statements and their possible realization patterns extracted from frames. Each instance is annotated with the three first letters of the core frame element it has been associated with.

examples (1) and (2). Intransitive usages, example (4). Semantic focus shifts, examples (3) and (5). Lexical variations and realizations of the same property, examples (1), (2) and (3).

#### 4 Discussion and related work

Applying frame semantics theory has been suggested before in the context of multilingual language generation (De Bleecker, 2005; Stede, 1996). However, to our knowledge, no generation application has tried to extract semantic frame information directly from a framenet resource and integrate the extracted information in the generation machinery. Perhaps because it is not until now that automatic processing of multilingual framenet data become available (Boas, 2009). Moreover, the rapid increase of Web ontologies has only recently become acknowledged in the NLG community, who started to recognize the new needs for establishing feasible methods that facilitate generation and aggregation of natural language from these emerging standards (Mellish and Sun, 2006).

Authors who have been experimenting with NLG from Web ontologies (Bontcheva and Wilks, 2004; Wilcock and Jokinen, 2003) have demonstrated the usefulness of performing aggregation and applying some kind of discourse structures in the early stages of the microplanning process. As mentioned in Section 1.1, peripheral elements can help in deciding on how the domain information should be packed into sentences. In the next step of our work, when we proceed with aggregations and discourse generation we intend to utilize the essential information provided by these elements.

Currently, the ontology properties are lexicalized manually, a process which relies solely on the frames and the ontology class hierarchies. To increase efficiency and accuracy, additional lexical

resources such as WordNet must be integrated into the system. This kind of integration has already proved feasible in the context of NLG (Jing and McKeown, 1998) and has several implications for automatic lexicalization.

#### 5 Conclusions

In this paper we presented on-going research on applying semantic frame theory to automate natural language template generation.

The proposed method has many advantages. First, the extracted templates and syntactic alternations provide varying degrees of complexity of linguistic entities which eliminate the need for manual input of language-specific heuristics. Second, the division of phases and the separation of the different tasks enables flexibility and re-use possibilities. This is in particular appealing for modular NLG systems. Third, it provides multilingual extension possibilities. Framenet resources offer an extended amount of semantic and syntactic phrase specifications that are only now becoming available in languages other than English. Because non-English framenets share the same type of conceptual backbone as the English FN, the steps involved in adapting the proposed method to other languages mainly concern lexicalization of the ontology properties.

Future work aims to enhance the proposed method along the lines discussed in Section 4 and test it on the Italian and Spanish framenets. We intend to experiment with the information about synonymous words and related terms provided in FN (which we haven't taken advantage of yet) and demonstrate how existing NLG applications that are designed to accommodate different user needs can benefit from it.

## Acknowledgments

The author would like to express her gratitude to Maria Toporowska Gronostaj for useful discussions about lexical semantics and to Olga Caprotti for making suggestions for improving the paper. I thank three anonymous reviewers for their encouraging comments on an earlier version of this paper.

## References

- Tim Berners-Lee. 2004. OWL Web Ontology Language reference, February. W3C Recommendation.
- Hans C. Boas. 2009. *Multilingual FrameNets in Computational Lexicography*.
- Kalina Bontcheva and Yorick Wilks. 2004. Automatic report generation from ontologies: the MIAKT approach. In *Proceedings of the Ninth International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 324–335.
- Stephan Busemann and Helmut Horacek. 1998. A flexible shallow approach to text generation. In *Proceedings of the 9th International Workshop on Natural Language Generation (IWNLG 98)*, pages 238–247, Niagara-on-the-Lake, Ontario.
- Inge M. R. De Bleecker. 2005. Towards an optimal lexicalization in a natural-sounding portable natural language generator for dialog systems. In *ACL '05: Proceedings of the ACL Student Research Workshop*, pages 61–66, Morristown, NJ, USA. Association for Computational Linguistics.
- Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to framenet. *International Journal of Lexicography*, 16(3):235–250.
- Dimitrios Galanis and Ion Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *Proceedings of the 11th European Workshop on Natural Language Generation, Schloss Dagstuhl*.
- Sabine Geldof and Walter van de Velde. 1997. An architecture for template-based (hyper)text generation. In *Proceedings of the Sixth European Workshop on Natural Language Generation*, pages 28–37, Duisburg, Germany.
- Amy Isard, Jon Oberlander, Ion Androutsopoulos, and Colin Matheson. 2003. Speaking the users' languages. *IEEE Intelligent Systems Magazine*, 18(1):40–45.
- Hongyan Jing and Kathleen McKeown. 1998. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *Proceedings of the 17th international conference on Computational linguistics*, pages 607–613, Morristown, NJ, USA. Association for Computational Linguistics.
- Chris Mellish and Xiantang Sun. 2006. The semantic web as a linguistic resource: Opportunities for natural language generation. *Knowledge-Based Systems*, 19(5):298–303.
- Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. MIT Press and The McGraw-Hill Companies, Inc.
- Ehud Reiter and Chris Mellish. 1993. Optimizing the costs and benefits of natural language generation. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 1164–1169, Chambery, France.
- Ehud Reiter. 1999. Shallow vs. deep techniques for handling linguistic constraints and optimisations. In DFKI, editor, *In Proceedings of the KI99 Workshop*.
- Manfred Stede. 1996. *Lexical semantics and knowledge representation in multilingual sentence generation*. Ph.D. thesis, Department of Computer Science, University of Toronto.
- Graham Wilcock and Kristiina Jokinen. 2003. Generating responses and explanations from RDF/XML and DAML+OIL. In *Knowledge and Reasoning in Practical Dialogue Systems IJCAI*, pages 58–63, Acapulco.





# ‘If you’ve heard it, you can say it’ - Towards an Account of Expressibility

**David D. McDonald**  
Raytheon BBN Technologies  
Cambridge, MA USA  
dmcdonald@bbn.com

**Charles F. Greenbacker**  
University of Delaware  
Newark, DE, USA  
charlieg@cis.udel.edu

## Abstract

We have begun a project to automatically create the lexico-syntactic resources for a microplanner as a side-effect of running a domain-specific language understanding system. The resources are parameterized synchronous TAG Derivation Trees. Since the KB is assembled from the information in the texts that these resources are abstracted from, it will decompose along those same lines when used for generation. As all possible ways of expressing each concept are pre-organized into general patterns known to be linguistically-valid (they were observed in natural text), we obtain an architectural account for expressibility.

## 1. Expressibility

People speak grammatically. They may stutter, restart, or make the occasional speech error, but all in all they are faithful to the grammar of the language dialects they use. One of the ways that a language generation system can account for this is through the use of grammar that defines all of the possible lexico-syntactic elements from which a text can be constructed and defines all their rules of composition, such as lexicalized Tree Adjoining Grammar (TAG). Without the ability to even formulate an ungrammatical text, such a generator provides an account for human grammaticality based on its architecture rather than its programmer.

We propose a similar kind of accounting for the problem of expressibility: one based on architecture rather than accident. *Expressibility*, as defined by Meteer (1992), is an issue for microplanners as they decide on which lexical and syntactic resources to employ. Not all of the options they might want to use are available in the language – they are not expressible. Consider the examples in Figure 1, adapted from Meteer 1992 pg. 50.

Expression	Construction (‘decide’)
“ <i>quick decision</i> ”	<result> + <quick>
“ <i>decide quickly</i> ”	<action> + <quick>
“ <i>important decision</i> ”	<result> + <important>
* “ <i>decide importantly</i> ”	<action> + <important>

**Figure 1: Constraints on expressibility: To say that there was a decision and it was important, you are forced to use the noun form because there is no adverbial form for *important* as there is for *quick***

In this short paper, we discuss our approach to expressibility. We describe in detail our novel method centered on how to use parser observations to guide generator decisions, and we provide a snapshot of the current status of our system implementation.

## 2. Related Work

Natural language generation (NLG) systems must have some way of making sure that the messages they build are actually expressible. Template-based generators avoid problems with expressibility largely by anticipating all of the wording that will be needed and packaging it in chunks that are guaranteed to compose correctly. Becker (2006), for example, does this via fully lexicalized TAG trees.

Among more general-purpose generators, one approach to expressibility is to look ahead into the lexicon, avoiding constructions that are lexically incompatible. Look-ahead is expensive, however, and is only practical at small abstraction distances such as Shaw’s re-writing sentence planner (1998).

Meteer’s own approach to expressibility started by interposing another level of representation between the microplanner and the surface realizer, an ‘abstract syntactic representation’ in the sense of RAGS (Cahill et al. 1999), that employed functional relationships (head, argument, matrix, adjunct) over semantically typed,

lexicalized constituents. This blocks *\*decide importantly* because ‘important’ only has a realization as a property and her composition rules prohibit using a property to modify an action (‘decide’). Shifting the perspective from the action to its result allows the composition to go through.

We are in sympathy with this approach – a microplanner needs its own representational level to serve as a scratch pad (if using a revision-based approach) or just as a scaffold to hold intermediate results. However, Meteor’s semantic and lexical constraints do require operating with fine-grain details. We believe that we can work with larger chunks that have already been vetted for expressibility because we’ve observed someone use them, either in writing or speech.

### 3. Method

Our approach is similar to that of Zhong & Stent (2005) in that we use the analysis of a corpus as the basis for creating the resources for the realization component. Several differences stand out. For one, we are working in specific domains rather than generic corpora like the WSJ. This enables the biggest difference: our analysis is performed by a completely accurate,<sup>1</sup> domain-specific NLU system (‘parser’)<sup>2</sup> based on a semantic grammar (McDonald 1993). It is reading for the benefit of a knowledge base, adding specific facts within instances of a highly structured, predefined prototypes. Such instances are used as the starting point for the generation process.

On the KB side, our present focus happens to be on hurricanes and the process they go through as they evolve. We have developed a semantic grammar for this domain, and it lets us analyze texts like these:<sup>3</sup>

- (1) “*Later that day it made landfall near the Haitian town of Jacmel.*”

<sup>1</sup> Parse accuracy and correct word sense interpretation is only possible if the semantic domain under analysis is restricted by topic and sublanguage.

<sup>2</sup> Most systems referred to as “parsers” stop at a structural description. Ours stops at the level of a disambiguated conceptual model and is more integrated than most.

<sup>3</sup> #1 and 2 are from the Wikipedia article on Hurricane Gustav. #3 is from a New York Times article.

- (2) “*... and remained at that intensity until landfall on the morning of September 1 near Cocodrie, Louisiana.*”

- (3) “*By landfall on Monday morning ...*”

Such texts tell us how people talk about hurricanes, specifically here about landfall events. They tell us what combinations of entities are reasonable to include within single clauses (intensity, time, location), and they tell us which particular realizations of the landfall concept have been used in which larger linguistic contexts. They also indicate what information can be left out under the discourse conditions defined by the larger texts they appear in.<sup>4</sup>

As different texts are read, we accumulate different realization forms for the same content. In example #1, landfall is expressed via the idiom *make landfall*, the time is given in an initial adverbial, and the location as a trailing adjunct. In #2, the landfall stands by itself as the head of a time-adverbial and the time and location are adjuncts off of it. This set of alternative phrasings provides the raw material for the microplanner to work with – a natural set of paraphrases.

#### 3.1 Derivation Trees as templates

As shown in Figure 3, to create resources for the microplanner, we start with the semantic analysis that the parser anchors to its referent when it instantiates the appropriate event type within the prototypical model of what hurricanes do, here a ‘landfall event’, noting the specific time and location. Following Bateman (e.g. 2007) and Meteor (1992), we work with typed, structured objects organized under a foundational ontology.<sup>5</sup> Figure 2 shows the current definition of the landfall class in a local notation for OWL Full.

```
(Class HurricaneLandfall
  (restrict hurricane - Hurricane)
  (restrict intensity - Saffir-Simpson)
  (restrict location - PhysEndurant)
  (restrict time - Date&Time))
```

Figure 2. The Landfall class

<sup>4</sup> For example, in #1 and #3 the precise date had been given already in earlier sentences.

<sup>5</sup> An extension of Dolce (Gangemi et al. 2002).

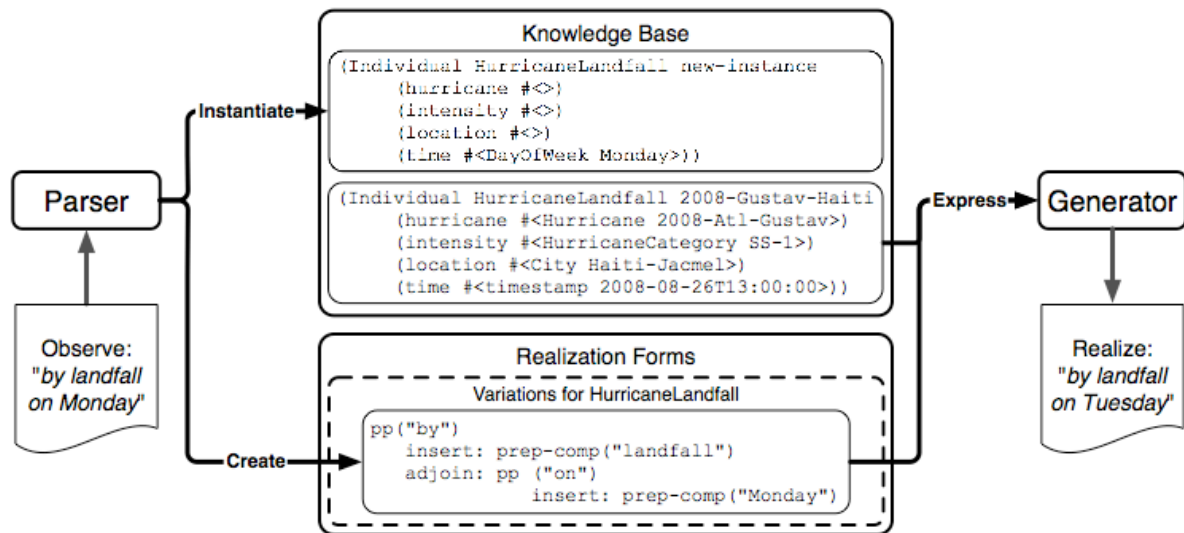


Figure 3. Overview

The semantic analysis recursively maps constituents' referents to properties of a class instance. Accompanying it is a syntactic analysis in the form of a TAG Derivation Tree<sup>6</sup> (DT) where each of its nodes (initial trees, insertions or adjunctions) points both to its lexical anchor and its specific correspondence in the domain model.

To create a reusable resource, we abstract away from the lexicalization in these DT/model-anchored pairs, and replace it with the corresponding model classes as determined by the restrictions on the properties. For example, the day of the week in #3, lexically given as *Monday morning* and then dereferenced to an object with the meaning '9/1/2008 before noon' is replaced in the resource with that object's type.

The result is a set of templates associated with the combination of types that corresponds to the participants in its source text – the more composed the type, the more insertions / adjunctions in the template derivation tree.

### 3.2 Synchronous TAGS

This combination of derived trees and model-levels classes and properties where the nodes of the two structures are linked is a *synchronous TAG* (ST). As observed by Shieber and Schabes (1991) who introduced this notion, “[STs] make the fine-grained correspondences between expressions of natural language and their meanings explicit by ... node linking”.

<sup>6</sup> The primary analysis is phrase structure in a chart, but since every rule in the grammar corresponds to either a lexicalized insertion or adjunction, the pattern of rule application is read out as a TAG derivation tree.

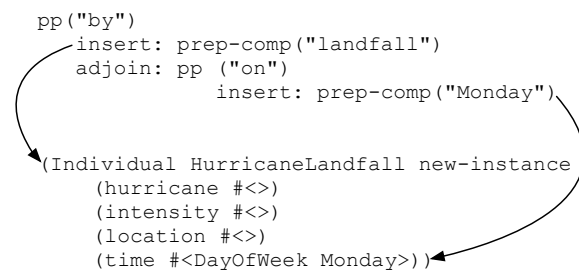


Figure 4. Synchronous TAG

In particular, they observe that STs solve an otherwise arbitrary problem of ‘where does one start’ when faced with a bag of content to be realized as a text. Our STs identify natural ‘slices’ of the content – those parts that have already been observed to have been realized together in a naturally occurring text.

Because we have the luxury to be creating the knowledge base of our hurricane model by the accretion of relationships among individually small chunks of information (a triple store), we can take synchronous TAGS a step further and allow them to dictate the permitted ways that information can be delimited within the KB for purposes of generation following the ideas in (Stone 2002).

If we can surmount the issues described below, this structure – that one can only select for generation units of content of the types that have been observed to be used together (the model side of the STs) – is a clean architectural explanation of how it is that the generator's messages are always expressible.

## 4. State of Development

We are at an early stage in our work. Everything we have described is implemented, but only on a

‘thin slice’ to establish that our ideas were credible. There are many issues to work out as we ‘bulk up’ the system and begin to actually integrate it in a ‘tactical’ microplanner and begin to actually do the style of macro-planning (determining the relevant portions of the domain model to use as content given the intent and affect) that our use of synchronous TAGS should allow. The most salient issues are how broadly we should generalize when we substitute domain types for lexicalizations in the templates, and what contextual information must be kept with the templates.

The type generalizations need to be broad enough to encompass as many substitutions as possible, while being strict enough to ensure that when the template is applied to those objects the realizations available to them permit them to be expressed in that linguistic context.<sup>7</sup>

The examples all have specific contexts in the sentences and recent discourse. Two of them (#2, #3) are using the landfall event as a time phrase. Can we move them and still retain the naturalness of the original (e.g. from sentence initial to sentence final), or does this sort of information need to be encoded?

Another issue is how to evaluate a system like this. Given the accuracy of the analysis, recreating the source text is trivial, so comparison to the source of the resources as a gold standard is meaningless. Some alternative must be found.

While we work out these issues, we are extending the NLU domain model and grammar to cover more cases and thence create more synchronized TAG templates. We then manually identify alternative domain content to apply to them in order to explore the space of realizations and identify unforeseen interactions.

Our short-term goals are to vastly increase the grammar coverage for our motivating examples and to hand over all microplanning decisions to the system itself. Long-term goals include broadening the coverage further still, to as open a domain as is feasible, as well as testing different macroplanners and applications with which to drive the entire process. Among several possibilities are automatic merged-and-modified summarization and a query-based discourse system.

<sup>7</sup> In our example, substituting different days and times is obvious (*by landfall on the afternoon of August 22*), but as we move away from that precise set of types (general-time-of-day + date) we see that what had been lexically fixed in the derivation tree (*by landfall on*) has to shift: ... *at 2:00 on August 22*.

## 5. Discussion

Because the phrasal patterns observed in the corpus act as templates guiding the generation process, and as the underlying NLU system and generator (McDonald 1993, Meteer et al. 1987) are mature and grounded in linguistic principles, our system combines template-based and theory-based approaches.

Van Deemter et al. (2005) outlined three criteria for judging template-driven applications against “standard” (non-template) NLG systems. (1) *Maintainability* is addressed by the fact that our templates aren’t hand-made. To extend the set of available realization forms we expose the NLU system to more text. The subject domain has to be one that has already been modeled, but we are operating from the premise that a NLG component would only bother to speak about things that the system as a whole understands. (2) *Output quality and variability* are determined by the corpus; using corpora containing high quality and varied constructions will enable similar output from the generator. (3) Most crucially, our parser and generator components are linguistically *well-founded*. Composition into our ‘templates’ is smoothly accommodated (extra modifiers, shifts in tense or aspect, application of transformations over the DT to form questions, relative clauses, dropped constituents under conjunction). The fully-articulated syntactic structure can be automatically annotated to facilitate prosody or to take information structure markup on the DT.

The closest system to ours may be Marciniak & Strube (2005) who also use an annotated corpus as a knowledge source for generation, getting their annotations via “a simple rule-based system tuned to the given types of text”. As far as we can tell, they are more concerned with discourse while we focus on the integration with the underlying knowledge base and how that KB is extended over time.

Like them, we believe that one of the most promising aspects of this work going forward is that the use of a parser provides us with “self-labeling data” to draw on for statistical analysis. Such training material would reduce the effort required to adapt a generator to a new domain, while simultaneously improving its output.

## Acknowledgments

This work was supported in part by the BBN POIROT project: DARPA IPTO contract FA865 0-06-C-7606.

## References

- John Bateman, Thora Tenbrink, and Scott Farrar. 2007. The Role of Conceptual and Linguistic Ontologies in Interpreting Spatial Discourse. *Discourse Processes*, 44(3):175–213.
- Tilman Becker. 2006. Natural Language Generation with Fully Specified Templates. In W. Wahlster (Ed.), *SmartKom: Foundations of Multimodal Dialog Systems*, 401–410. Springer, Berlin Heidelberg.
- Lynne Cahill, Christy Doran, Roger Evans, Chris Mellish, Daniel Paiva, Mike Reape, Donia Scott, & Neil Tipper. 1999. *Towards a Reference Architecture for Natural Language Generation Systems, The RAGS project*. ITRI technical report number ITRI-99-14, University of Brighton, March.
- Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, & Luc Schneider. 2002. Sweetening Ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Acquisition, Modeling and Management (EKAW)*, pages 166–181, Sigüenza, Spain, October 1–4.
- Tomasz Marciniak & Michael Strube. 2005. Using an Annotated Corpus As a Knowledge Source For Language Generation. In *Proceedings of the Corpus Linguistics 2005 Workshop on Using Corpora for Natural Language Generation (UCNLG)*, pages 19–24, Birmingham, UK, July 14.
- David McDonald. 2003. The Interplay of Syntactic and Semantic Node Labels in Partial Parsing, in the proceedings of the Third International Workshop on Parsing Technologies, August 10-13, 1993 Tilburg, The Netherlands, pp. 171-186; revised version in Bunt and Tomita (eds), *Recent Advances in Parsing Technology*, Kluwer Academic Publishers, pgs. 295-323.
- Marie W. Meteer. 1992. *Expressibility and the Problem of Efficient Text Planning*. Pinter, London.
- Marie Meteer, David McDonald, Scott Anderson, David Forster, Linda Gay, Alison Huettner & Penelope Sibun. 1987. Mumble-86: Design and Implementation, TR #87-87 Dept. Computer & Information Science, UMass., September 1987, 174 pgs.
- James Shaw. 1998. Clause Aggregation Using Linguistic Knowledge. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 138–147, Niagara-on-the-Lake, Ontario, August 5–7.
- Stuart Shieber & Yves Schabes. 1991. Generation and synchronous tree-adjointing grammar. *Computational Intelligence*, 7(4):220–228.
- Matthew Stone. 2003. Specifying Generation of Referring Expressions by Example. In *Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, pages 133–140, Stanford, March.
- Kees van Deemter, Emiel Krahmer, & Mariët Theune. 2005. Real versus Template-Based Natural Language Generation: A False Opposition? *Computational Linguistics*, 31(1):15–24.
- Huvava Zhong & Amada Stent. 2005. Building Surface Realizers Automatically From Corpora. In *Proceedings of the Corpus Linguistics 2005 Workshop on Using Corpora for Natural Language Generation (UCNLG)*, pages 49–54, Birmingham, UK, July 14.



# Cross-Linguistic Attribute Selection for REG: Comparing Dutch and English

**Mariët Theune**

University of Twente  
The Netherlands

M.Theune@utwente.nl

**Ruud Koolen**

Tilburg University  
The Netherlands

R.M.F.Koolen@uvt.nl

**Emiel Krahmer**

Tilburg University  
The Netherlands

E.J.Krahmer@uvt.nl

## Abstract

In this paper we describe a cross-linguistic experiment in attribute selection for referring expression generation. We used a graph-based attribute selection algorithm that was trained and cross-evaluated on English and Dutch data. The results indicate that attribute selection can be done in a largely language independent way.

## 1 Introduction

A key task in natural language generation is referring expression generation (REG). Most work on REG is aimed at producing distinguishing descriptions: descriptions that uniquely characterize a target object in a visual scene (e.g., “the red sofa”), and do not apply to any of the other objects in the scene (the distractors). The first step in generating such descriptions is attribute selection: choosing a number of attributes that uniquely characterize the target object. In the next step, realization, the selected attributes are expressed in natural language. Here we focus on the attribute selection step. We investigate to which extent attribute selection can be done in a language independent way; that is, we aim to find out if attribute selection algorithms trained on data from one language can be successfully applied to another language. The languages we investigate are English and Dutch.

Many REG algorithms require training data, before they can successfully be applied to generate references in a particular domain. The Incremental Algorithm (Dale and Reiter, 1995), for example, assumes that certain attributes are more preferred than others, and it is assumed that determining the preference order of attributes is an empirical matter that needs to be settled for each new domain. The graph-based algorithm (Krahmer et al., 2003), to give a second example, similarly assumes that certain attributes are preferred (are

“cheaper”) than others, and that data are required to compute the attribute-cost functions.

Traditional text corpora have been argued to be of restricted value for REG, since these typically are not “semantically transparent” (van Deemter et al., 2006). Rather what seems to be needed is data collected from human participants, who produce referring expressions for specific targets in settings where all properties of the target *and* its distractors are known. Needless to say, collecting and annotating such data takes a lot of time and effort. So what to do if one wants to develop a REG algorithm for a new language? Would this require a new data collection, or could existing data collected for a *different* language be used? Clearly, linguistic realization is language dependent, but to what extent is attribute selection language dependent? This is the question addressed in this paper.

Below we describe the English and Dutch corpora used in our experiments (Section 2), the graph-based algorithm we used for attribute selection (Section 3), and the corpus-based attribute costs and orders used by the algorithm (Section 4). We present the results of our cross-linguistic attribute selection experiments (Section 5) and end with a discussion and conclusions (Section 6).

## 2 Corpora

### 2.1 English: the TUNA Corpus

For English data, we used the TUNA corpus of object descriptions (Gatt et al., 2007). This corpus was created by presenting the participants in an on-line experiment with a visual scene consisting of seven objects and asking them to describe one of the objects, the target, in such a way that it could be uniquely identified. There were two experimental conditions: in the +LOC condition, the participants were free to describe the target object using any of its properties, including its location on the screen, whereas in the -LOC condition they

were discouraged (but not prevented) from mentioning object locations. The resulting object descriptions were annotated using XML and combined with an XML representation of the visual scene, listing all objects and their properties in terms of attribute-value pairs. The TUNA corpus is split into two domains: one with descriptions of furniture and one with descriptions of people.

The TUNA corpus was used for the comparative evaluation of REG systems in the TUNA Challenges (2007-2009). For our current experiments, we used the TUNA 2008 Challenge training and development sets (Gatt et al., 2008) to train and evaluate the graph-based algorithm on.

## 2.2 Dutch: the D-TUNA Corpus

For Dutch, we used the D(utch)-TUNA corpus of object descriptions (Koolen and Kraemer, 2010). The collection of this corpus was inspired by the TUNA experiment described above, and was done using the same visual scenes. There were three conditions: text, speech and face-to-face. The text condition was a replication (in Dutch) of the TUNA experiment: participants typed identifying descriptions of target referents, distinguishing them from distractor objects in the scene. In the other two conditions participants produced spoken descriptions for an addressee, who was either visible to the speaker (face-to-face condition) or not (speech condition). The resulting descriptions were annotated semantically using the XML annotation scheme of the English TUNA corpus.

The procedure in the D-TUNA experiment differed from that used in the original TUNA experiment in two ways. First, the D-TUNA experiment used a laboratory-based set-up, whereas the TUNA study was conducted on-line in a relatively uncontrolled setting. Second, participants in the D-TUNA experiment were completely prevented from mentioning object locations.

## 3 Graph-Based Attribute Selection

For attribute selection, we use the graph-based algorithm of Kraemer et al. (2003), one of the highest scoring attribute selection methods in the TUNA 2008 Challenge (Gatt et al. (2008), table 11). In this approach, a visual scene with target and distractor objects is represented as a labelled directed graph, in which the objects are modelled as nodes and their properties as looping edges on the corresponding nodes. To select the

attributes for a distinguishing description, the algorithm searches for a subgraph of the scene graph that uniquely refers to the target referent. Starting from the node representing the target, it performs a depth-first search over the edges connected to the subgraph found so far. The algorithm's output is the cheapest distinguishing subgraph, given a particular *cost function* that assigns costs to attributes.

By assigning zero costs to some attributes, e.g., the type of an object, the human tendency to mention redundant attributes can be mimicked. However, as shown by Viethen et al. (2008), merely assigning zero costs to an attribute is not a sufficient condition for inclusion; if the graph search terminates before the free attributes are tried, they will not be included. Therefore, the order in which attributes are tried must be explicitly controlled.

Thus, when using the graph-based algorithm for attribute selection, two things must be specified: (1) the cost function, and (2) the order in which the attributes should be searched. Both can be based on corpus data, as described in the next section.

## 4 Costs and Orders

For our experiments, we used the graph-based attribute selection algorithm with two types of cost functions: Stochastic costs and Free-Naïve costs. Both reflect (to a different extent) the relative attribute frequencies found in a training corpus: the more frequently an attribute occurs in the training data, the cheaper it is in the cost functions.

Stochastic costs are directly based on the attribute frequencies in the training corpus. They are derived by rounding  $-\log_2(P(v))$  to the first decimal and multiplying by 10, where  $P(v)$  is the probability that attribute  $v$  occurs in a description, given that the target object actually has this property. The probability  $P(v)$  is estimated by determining the frequency of each attribute in the training corpus, relative to the number of target objects that possess this attribute. Free-Naïve costs more coarsely reflect the corpus frequencies: very frequent attributes are “free” (cost 0), somewhat frequent attributes have cost 1 and infrequent attributes have cost 2. Both types of cost functions are used in combination with a stochastic ordering, where attributes are tried in the order of increasing stochastic costs.

In total, four cost functions were derived from the English corpus data and four cost functions derived from the Dutch corpus data. For each lan-



guage, we had two Stochastic cost functions (one for the furniture domain and one for the people domain), and two Free-Naïve cost functions (idem), giving eight different cost functions in total. For each language we determined two attribute orders to be used with the cost functions: one for the furniture domain and one for the people domain.

#### 4.1 English Costs and Order

For English, we used the Stochastic and Free-Naïve cost functions and the stochastic order from Kraemer et al. (2008). The Stochastic costs and order were derived from the attribute frequencies in the combined training and development sets of the TUNA 2008 Challenge (Gatt et al., 2008), containing 399 items in the furniture domain and 342 items in the people domain. The Free-Naïve costs are simplified versions of the stochastic costs. “Free” attributes are TYPE in both domains, COLOUR for the furniture domain and HASBEARD and HASGLASSES for the people domain. Expensive attributes (cost 2) are X- and Y-DIMENSION in the furniture domain and HAS-SUIT, HASSHIRT and HASTIE in the people domain. All other attributes have cost 1.

#### 4.2 Dutch Costs and Order

The Dutch Stochastic costs and order were derived from the attribute frequencies in a set of 160 items (for both furniture and people) randomly selected from the text condition in the D-TUNA corpus. Interestingly, our Stochastic cost computation method led to an assignment of 0 costs to the COLOUR attribute in the furniture domain, thus enabling the Dutch Stochastic cost function to include colour as a redundant property in the generated descriptions. In the English stochastic costs, none of the attributes are free. Another difference is that in the furniture domain, the Dutch stochastic costs for ORIENTATION attributes are much lower than the English costs (except with value FRONT); in the people domain, the same holds for attributes such as HASSUIT and HASTIE. These cost differences, which are largely reflected in the Dutch Free-Naïve costs, do not seem to be caused by differences in expressibility, i.e., the ease with which the attributes can be expressed in the two languages (Koolen et al., 2010); rather, they may be due to the fact that the human descriptions in D-TUNA do not include any DIMENSION attributes.

Language		Furniture		People	
Training	Test	Dice	Acc.	Dice	Acc.
Dutch	Dutch	0.92	0.63	0.78	0.28
	English	0.83	0.55	0.73	0.29
English	Dutch	0.87	0.58	0.75	0.25
	English	0.67	0.29	0.67	0.24

Table 1: Evaluation results for stochastic costs.

Language		Furniture		People	
Training	Test	Dice	Acc.	Dice	Acc.
Dutch	Dutch	0.94	0.70	0.78	0.28
	English	0.83	0.55	0.73	0.29
English	Dutch	0.94	0.70	0.78	0.28
	English	0.83	0.55	0.73	0.29

Table 2: Evaluation results for Free-Naïve costs.

## 5 Results

All cost functions were applied to both Dutch and English test data. As Dutch test data, we used a set of 40 furniture items and a set of 40 people items, randomly selected from the text condition in the D-TUNA corpus. These items had not been used for training the Dutch cost functions. As English test data, we used a subset of the TUNA 2008 development set (Gatt et al., 2008). To make the English test data comparable to the Dutch ones, we only included items from the -LOC condition (see Section 2.1). This resulted in 38 test items for the furniture domain, and 38 for the people domain.

Tables 1 and 2 show the results of applying the Dutch and English cost functions (with Dutch and English attribute orders respectively) to the Dutch and English test data. The evaluation metrics used, Dice and Accuracy (Acc.), both evaluate human-likeness by comparing the automatically selected attribute sets to those in the human test data. Dice is a set-comparison metric ranging between 0 and 1, where 1 indicates a perfect match between sets. Accuracy is the proportion of system outputs that exactly match the corresponding human data. The results were computed using the ‘teval’ evaluation tool provided to participants in the TUNA 2008 Challenge (Gatt et al., 2008).

To determine significance, we applied repeated measures analyses of variance (ANOVA) to the evaluation results, with three within factors: *training language* (Dutch or English), *cost function* (Stochastic or Free-Naïve), and *domain* (furniture or people), and one between factor representing *test language* (Dutch or English).

An overall effect of cost function shows that the Free-Naïve cost functions generally perform better

than the Stochastic cost functions (Dice:  $F(1,76) = 34.853$ ,  $p < .001$ ; Accuracy:  $F(1,76) = 13.052$ ,  $p = .001$ ). Therefore, in the remainder of this section we mainly focus on the results for the Free-Naïve cost functions (Table 2).

As can be clearly seen in Table 2, Dutch and English Free-Naïve cost functions give almost the same scores in both the furniture and the people domain, when applied to the same test language. The English Free-Naïve cost function performs slightly better than the Dutch one on the Dutch people data, but this difference is not significant.

An overall effect of test language shows that the cost functions (both Stochastic and Free-Naïve) generally give better Dice results on the Dutch data than for the English data (Dice:  $F(1,76) = 7.797$ ,  $p = .007$ ). In line with this, a two-way interaction between test language and training language (Dice:  $F(1,76) = 6.870$ ,  $p = .011$ ) shows that both the Dutch and the English cost functions perform better on the Dutch data than on the English data. However, the overall effect of test language did not reach significance for Accuracy, presumably due to the fact that the Accuracy scores on the English people data are slightly higher than those on the Dutch people data.

Finally, the cost functions generally perform better in the furniture domain than in the people domain (Dice:  $F(1,76) = 10.877$ ,  $p = .001$ ; Accuracy:  $F(1,76) = 16.629$ ,  $p < .001$ ).

## 6 Discussion

The results of our cross-linguistic attribute selection experiments show that Free-Naïve cost functions, which only roughly reflect the attribute frequencies in the training corpus, have an overall better performance than Stochastic cost functions, which are directly based on the attribute frequencies. This holds across the two languages we investigated, and corresponds with the findings of Kraemer et al. (2008), who compared Stochastic and Free-Naïve functions that were trained and evaluated on English data only. The difference in performance is probably due to the fact that Free-Naïve costs are less sensitive to the specifics of the training data (and are therefore more generally applicable) and do a better job of mimicking the human tendency towards redundancy.

Moreover, we found that Free-Naïve cost functions trained on different languages (English or Dutch) performed equally well when tested on the

same data (English or Dutch), in both the furniture and people domain. This suggests that attribute selection can in fact be done in a language independent way, using cost functions that have been derived from corpus data in one language to perform attribute selection for another language.

Our results did show an effect of test language on performance: both English and Dutch cost functions performed better when tested on the Dutch D-TUNA data than on the English TUNA data. However, this difference does not seem to be caused by language-specific factors but rather by the quality of the respective test sets. Although the English test data were restricted to the -LOC condition, in which using DIMENSION attributes was discouraged, still more than 25% of the English test data (both furniture and people) included one or more DIMENSION attributes, which were never selected for inclusion by either the English or the Dutch Free-Naïve cost functions. The Dutch test data, on the other hand, did not include any DIMENSION attributes. In addition, the English test data contained more non-unique descriptions of target objects than the Dutch data, in particular in the furniture domain. These differences may be due to the fact that data collection was done in a more controlled setting for D-TUNA than for TUNA. In other words, the seeming effect of test language does not contradict our main conclusion that attribute selection is largely language independent, at least for English and Dutch.

The success of our cross-linguistic experiments may have to do with the fact that English and Dutch hardly differ in the expressibility of object attributes (Koolen et al., 2010). To determine the full extent to which attribute selection can be done in a language-dependent way, additional experiments with less similar languages are necessary.

## Acknowledgements

We thank the TUNA Challenge organizers for the English data and the evaluation tool used in our experiments; Martijn Goudbeek for helping with the statistical analysis; and Pascal Tousek, Ivo Brugman, Jette Viethen, and Iris Hendrickx for their contributions to the graph-based algorithm. This research is part of the VICI project ‘Bridging the gap between psycholinguistics and computational linguistics: the case of referring expressions’, funded by the Netherlands Organization for Scientific Research (NWO Grant 277-70-007).

## References

- R. Dale and E. Reiter. 1995. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- A. Gatt, I. van der Sluis, and K. van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007)*, pages 49–56.
- A. Gatt, A. Belz, and E. Kow. 2008. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 198–206.
- R. Koolen and E. Krahrmer. 2010. The D-TUNA corpus: A Dutch dataset for the evaluation of referring expression generation algorithms. In *Proceedings of the 7th international conference on Language Resources and Evaluation (LREC 2010)*.
- R. Koolen, A. Gatt, M. Goudbeek, and E. Krahrmer. 2010. Overspecification in referring expressions: Causal factors and language differences. Submitted.
- E. Krahrmer, S. van Erk, and A. Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- E. Krahrmer, M. Theune, J. Viethen, and I. Hendrickx. 2008. Graph: The costs of redundancy in referring expressions. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 227–229.
- K. van Deemter, I. I. van der Sluis, and A. Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 130–132.
- J. Viethen, R. Dale, E. Krahrmer, M. Theune, and P. Tousek. 2008. Controlling redundancy in referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 239–246.



# Grouping Axioms for More Coherent Ontology Descriptions

**Sandra Williams**

The Open University  
Milton Keynes, United Kingdom  
s.h.williams@open.ac.uk

**Richard Power**

The Open University  
Milton Keynes, United Kingdom  
r.power@open.ac.uk

## Abstract

Ontologies and datasets for the Semantic Web are encoded in OWL formalisms that are not easily comprehended by people. To make ontologies accessible to human domain experts, several research groups have developed ontology verbalisers using Natural Language Generation. In practice ontologies are usually composed of simple axioms, so that realising them separately is relatively easy; there remains however the problem of producing texts that are coherent and efficient. We describe in this paper some methods for producing sentences that aggregate over sets of axioms that share the same logical structure. Because these methods are based on logical structure rather than domain-specific concepts or language-specific syntax, they are generic both as regards domain and language.

## 1 Introduction

When the Semantic Web becomes established, people will want to build their own knowledge bases (i.e., ontologies, or TBox axioms, and data, or ABox axioms<sup>1</sup>). Building these requires a high level of expertise and is time-consuming, even with the help of graphical interface tools such as Protégé (Knublauch et al., 2004). Fortunately, natural language engineers have provided a solution to at least part of the problem: verbalisers, e.g., the OWL ACE verbaliser (Kaljurand and Fuchs, 2007).

Ontology verbalisers are NLG systems that generate controlled natural language from Semantic

<sup>1</sup>Description Logic (DL) underlies the Web Ontology Language OWL. DL distinguishes statements about classes (TBox) from those about individuals (ABox). OWL covers both kinds of statements, which in OWL terminology are called ‘axioms’.

Web languages, see Smart (2008). Typically they generate one sentence per axiom: for example, from the axiom<sup>2</sup>  $Cat \sqsubseteq Animal$  the OWL ACE verbaliser (Kaljurand and Fuchs, 2007) generates ‘Every cat is an animal’. The result is not a coherent text, however, but a disorganised list, often including inefficient repetitions such as:

Every cat is an animal.  
Every dog is an animal.  
Every horse is an animal.  
Every rabbit is an animal.

An obvious first step towards improved efficiency and coherence would be to replace such lists with a single aggregated sentence:

The following are kinds of animals: cats, dogs, horses and rabbits.

In this paper, we show how all axiom patterns in  $\mathcal{EL}++$ , a DL commonly used in the Semantic Web, can be aggregated *without further domain knowledge*, and describe a prototype system that performs such aggregations. Our method aggregates axioms *while they are still in logical form*, i.e., as part of sentence planning but before converting to a linguistic representation and realising as English sentences. This approach is somewhat different from that proposed by other researchers who convert ontology axioms to linguistic structures before aggregating (Hielkema, 2009; Galanis et al., 2009; Dongilli, 2008). We present results from testing our algorithm on over fifty ontologies from the Tones repository<sup>3</sup>.

## 2 Analysis of Axiom Groupings

In this section we analyse which kinds of axioms might be grouped together. Power (2010) anal-

<sup>2</sup>For brevity we use logic notation rather than e.g., OWL Functional Syntax: `subClassOf(class(ns:cat) class(ns:animal))` where `ns` is any valid namespace. The operator  $\sqsubseteq$  denotes the subclass relation,  $\sqcap$  denotes class intersection, and  $\exists P.C$  the class of individuals bearing the relation  $P$  to one or more members of class  $C$ .

<sup>3</sup><http://owl.cs.manchester.ac.uk/>

No.	Logic	OWL	%
1	$A \sqsubseteq B$	subClassOf(A B)	51
2	$A \sqsubseteq \exists P.B$	subClassOf(A someValuesFrom(P B))	33
3	$[a, b] \in P$	propertyAssertion(P a b)	8
4	$a \in A$	classAssertion(A a)	4

Table 1: The four most common axiom patterns.

used axiom patterns present in the same fifty ontologies. In spite of the richness of OWL, the surprising result was that *only four* relatively simple patterns dominated, accounting for 96% of all patterns found in more than 35,000 axioms. Overall there were few unique patterns, typically only 10 to 20, and up to 34 in an unusually complex ontology. Table 1 lists the common patterns in logic notation and OWL Functional Syntax, and also gives the frequencies across the fifty knowledge bases. Examples of English paraphrases for them are:

1. Every Siamese is a cat.
2. Every cat has as body part a tail.
3. Mary owns Kitty.
4. Kitty is a Siamese.

When two or more axioms conform to a pattern:

$$\begin{aligned} A &\sqsubseteq B \\ A &\sqsubseteq C \\ B &\sqsubseteq C \\ C &\sqsubseteq D \end{aligned}$$

there are two techniques with which to aggregate them: merging and chaining. If the right-hand sides are identical we can merge the left-hand sides, and vice versa:<sup>4</sup>

$$\begin{aligned} [A, B] &\sqsubseteq C \\ A &\sqsubseteq [B, C] \end{aligned}$$

Alternatively, where the right-hand side of an axiom is identical to the left-hand side of another axiom, we can ‘chain’ them:

$$A \sqsubseteq B \sqsubseteq C \sqsubseteq D$$

Merging compresses the information into a more efficient text, as shown in the introduction, while chaining orders the information to facilitate inference — for example, ‘Every A is a B and every B is a C’ makes it easier for readers to draw the inference that every A is a C.

<sup>4</sup>We regard expressions like  $A \sqsubseteq [B, C]$  and  $A \sqsubseteq B \sqsubseteq C$  as shorthand forms allowing us to compress several axioms into one formula. For merges one could also *refactor* the set of axioms into a new axiom: thus for example  $A \sqsubseteq [B, C]$  could be expressed as  $A \sqsubseteq (B \sqcap C)$ , or  $[A, B] \sqsubseteq C$  as  $(A \sqcup B) \sqsubseteq C$ . This formulation would have the advantage of staying within the normal notation and semantics of DL; however, it is applicable only to merges, not to chains.

	1.	2.	3.	4.
1.	L,R,C	×,R?,×	×,×,×	L?,×,C
2.	L,R,×	×,×,×	×,×,×	×,×,×
3.	L,R,×	×,×,×	L,R,×	×,R?,×
4.	L,R,×	×,×,×	×,×,×	L,R,×

Table 2: Aggregating common axioms: 1.  $A \sqsubseteq B$ , 2.  $A \sqsubseteq \exists P.B$ , 3.  $[a, b] \in P$ , 4.  $a \in A$ 

Table 2 summarises our conclusions on whether each pair of the four common patterns can be merged or chained. Each cell contains three entries, indicating the possibility of left-hand-side merge (L), right-hand-side merge (R), and chaining (C). As can be seen, some merges or chains are possible across different patterns, but the safest aggregations are those grouping axioms with the same pattern (down the diagonal), and it is these on which we focus here.

### 3 Merging Similar Patterns

Function	Merge Patterns
$f_1(A)$	$f_1([A_1, A_2, A_3, \dots])$
$f_2(A, B)$	$f_2([A_1, A_2, A_3, \dots], B)$ $f_2(A, [B_1, B_2, B_3, \dots])$
$f_3(A, B, C)$	$f_3([A_1, A_2, A_3, \dots], B, C)$ $f_3(A, [B_1, B_2, B_3, \dots], C)$ $f_3(A, B, [C_1, C_2, C_3, \dots])$

Table 3: Generic merging rules.

If we represent ABox and TBox axioms as Prolog terms (or equivalently in OWL Functional Syntax), they take the form of functions with a number of arguments — for example `subClassOf(A, B)`, where `subClassOf` is the functor, `A` is the first argument and `B` is the second argument. We can then formulate generic aggregation rules for merging one-, two- and three-argument axioms, as shown in table 3.

In general, we combine axioms for which the functor is the same and only one argument differs. We do not aggregate axiom functions with more than three arguments. The merged constituents must be different expressions with the same logical form.

### 4 Implementation

This section describes a Prolog application which performs a simple verbalisation including aggregation. It combines a generic grammar for realising logical forms with a domain-specific lexicon

derived from identifiers and labels within the input ontology.

Input to the application is an OWL/XML file.<sup>5</sup> Axioms that conform to  $\mathcal{EL}++$  DL are selected and converted into Prolog format. A draft lexicon is then built automatically from the identifier names and labels, on the assumption that classes are lexicalised by noun groups, properties by verb groups with valency two, and individuals by proper nouns.

Our aggregation rules are applied to axioms with the same logical form. The first step picks out all the logical patterns present in the input ontology by abstracting from atomic terms. The next step searches for all axioms matching each of the patterns present. Then within each pattern-set, the algorithm searches for axioms that differ by only one argument, grouping axioms together in the ways suggested in table 3. It exhaustively lists every possible grouping and builds a new, aggregated axiom placing the values for the merged argument in a list, e.g., consider the axioms:

```
subClassOf(class(cat), class(feline)).
subClassOf(class(cat), class(mammal)).
subClassOf(class(dog), class(mammal)).
subClassOf(class(mouse), class(mammal)).
```

Identical first arguments  $\implies$

```
subClassOf(class(cat),
           [class(feline),
            class(mammal)]).
‘Every cat is a feline and a mammal.’
```

Identical second arguments  $\implies$

```
subClassOf([class(cat), class(dog),
            class(mouse)], class(mammal)).
‘The following are kinds of mammal:
  cats, dogs and mice.’
```

For all axioms with an identical first argument, `class(cat)`, the algorithm places the second arguments in a list, `[class(feline), class(mammal)]`, and builds a new axiom with the first argument and the merged second argument. From this, our realiser generates the sentence ‘Every cat is a feline and a mammal.’ A similar process is performed on first arguments when the second arguments are identical.

To construct the grammar, we first formulated rules for realising single axioms, and then added rules for the aggregated patterns, incorporating aggregation cues such as ‘both’ and ‘the following:’ (Dalianis and Hovy, 1996). For the wording of single axioms we relied mainly on proposals

<sup>5</sup>We convert OWL to OWL/XML with the Manchester OWL Syntax Converter <http://owl.cs.manchester.ac.uk/converter/>

from the OWL Controlled Natural Language task force (Schwitter et al., 2008), so obtaining reasonably natural sentences for common axiom patterns, even though some less common axioms such as those describing attributes of properties (e.g., domain, range, functionality, reflexivity, transitivity) are hard to express without falling back on technical concepts from the logic of relations; for these we have (for now) allowed short technical formulations (e.g., ‘The property “has as part” is transitive’). With these limitations, the grammar currently realises any single axiom conforming to  $\mathcal{EL}++$ , or any aggregation of  $\mathcal{EL}++$  axioms through the merge rules described above. Table 4 lists example aggregated axiom patterns and English realisations generated with our grammar.

## 5 Testing the ‘Merging’ Algorithm

Unit	Original	Aggregated	Reduction
Sentences	35,542	11,948	66%
Words	320,603	264,461	18%

Table 5: Reduction achieved by aggregating

We have tested our generic merging rules on axioms conforming to  $\mathcal{EL}++$  in a sample of around 50 ontologies. Table 5 shows the reduction in the number of generated sentence after aggregation. Remember that previously, the system generated one sentence for every axiom (35,542 sentences), but with aggregation this is reduced to 11,948 sentences, an overall reduction of 66%. However, aggregation increases sentence length so the saving in words is only 18%.

The effect of merging is to replace a large number of short sentences with a smaller number of longer ones. Sometimes the aggregated sentences were very long indeed, e.g., when a travel ontology cited 800 instances of the class `island` — perhaps such cases would be expressed better by a table than by prose<sup>6</sup>.

The algorithm computes all possible merges, so we get, for instance, Fred described as a person in both ‘The following are people: Fred, ...’ and ‘Fred is all of the following: a person, ...’. This means that the greater efficiency achieved through aggregation may be counterbalanced by the extra text required when the same axiom participates in several merges — for a few of our ontologies, in

<sup>6</sup>In a summary one might instead simply give a count and an example: ‘There are 800 islands, e.g., The Isle of Skye’.

Aggregated Axiom Pattern	Example of Generated Text
subClassOf( $C_1, C_2, \dots$ ), $C_3$ ). subClassOf( $C_1, [C_2, C_3, \dots]$ ).	The following are kinds of vehicles: a bicycle, a car, a truck and a van. Every old lady is all of the following: a cat owner, an elderly and a woman.
subClassOf( $C_1, C_2, \dots$ ), objectSomeValuesFrom( $P_1, C_3$ ). subClassOf( $C_1, [objectSomeValuesFrom(P_1, C_2)$ objectSomeValuesFrom( $P_2, C_3$ ))].	The following are kinds of something that has as topping a tomato: a fungi, a fiorella and a margherita. Every fiorella is something that has as topping a mozzarella and is something that has as topping an olive.
classAssertion( $C_1, [I_1, I_2, \dots]$ ). classAssertion( $[C_1, C_2, \dots], I$ ).	The following are people: Fred, Joe, Kevin and Walt. Fred is all of the following: an animal, a cat owner and a person.
objectPropertyAssertion( $P_1, [I_1, I_2, I_3, I_4]$ ). objectPropertyAssertion( $P_1, I_4, [I_1, I_2, I_3]$ ).	The following are pet of Walt: Dewey, Huey and Louie. Walt has as pet Dewey, Huey and Louie.
disjointClasses( $[C_1, C_2, \dots], C_3$ ). disjointClasses( $C_1, [C_2, C_3, \dots]$ ).	None of the following are mad cows: an adult, ... a lorry or a lorry driver. No grownup is any of the following: a kid, a mad cow, a plant, or a tree.
dataPropertyDomain( $[P_1, P_2, \dots], C_1$ ).	If any of the following relationships hold between X and Y then X must be a contact: "has as city", "has as street" and "has as zip code".
dataPropertyRange( $[P_1, P_2, \dots], C_1$ ).	If any of the following relationships hold between X and Y then Y must be a string: "has as city", "has as e mail" and "has as street".
differentIndividuals( $I_1, [I_2, I_3, \dots]$ ). differentIndividuals( $[I_1, I_2, \dots], I_3$ ).	The One Star Rating is a different individual from any of the following: the Three Star Rating or the Two Star Rating.
equivalentDataProperties( $P_1, [P_2, P_3, \dots]$ ). equivalentDataProperties( $[P_1, P_2, \dots], P_3$ ). equivalentObjectProperties( $[P_1, P_2, \dots], P_3$ ).	The following properties are equivalent to the property "has as zip code": "has as post code", "has as zip" and "has as postcode". The following properties are equivalent to the property "has as father": ....
negativeObjectPropertyAssertion( $P_1, [I_1, I_2, \dots], I_3$ ). negativeObjectPropertyAssertion( $P_1, I_1, [I_2, I_3, \dots]$ ).	None of the following are pet of Walt: Fluffy, Mog or Rex. It is not true that Walt has as pet Fluffy or Rex.

Table 4: Example realisations of common aggregated  $\mathcal{EL}++$  axiom patterns.

fact, the word count for the aggregated version was *greater*. This is an interesting problem that we have not seen treated elsewhere. Merely pursuing brevity, one might argue that an axiom already included in a merge should be removed from any other merges in which it participates; on the other hand, the arbitrary exclusion of an axiom from a list might be regarded as misleading. For now we have allowed repetition, leaving the problem to future work.

## 6 Related Work

Reape and Mellish's (1999) survey of aggregation in NLG proposed a continuum of definitions ranging from narrow to wide. Our technique fits into the narrow definition, i.e., it is language-independent, operating on non-linguistic conceptual representations with the aim of minimising redundancy and repetition. It implements the subject and predicate grouping rules and aggregation cues suggested by Dalianis and Hovy (1996).

Recent NLG systems that aggregate data from ontologies (Hielkema, 2009; Galanis and Androutsopoulos, 2007; Dongilli, 2008) do not perform aggregation directly on axioms, but only *after* converting them to linguistic representations. Moreover, their systems generate only from ABox axioms in restricted domains while ours generates English for *both* ABox and TBox in *any domain*.

The approach most similar to ours is that of Bontcheva and Wilks (2004), who aggregate a subset of RDF triples after domain-dependent discourse structuring — a task equiv-

alent to merging axioms that conform to the `objectPropertyAssertion` pattern in table 4.

## 7 Conclusion

We have demonstrated that for the  $\mathcal{EL}++$  DL that underlies many Semantic Web ontologies we can define generic aggregation rules based on logical structure, each linked to a syntactic rule for expressing the aggregated axioms in English. The work described here is a first step in tackling a potentially complex area, and relies at present on several intuitive assumptions that need to be confirmed empirically. First, from an examination of all combinations of the four commonest axiom patterns, we concluded that axioms sharing the same pattern could be combined more effectively than axioms with different patterns, and therefore focussed first on same-pattern merges with variations in only one constituent. Secondly, after systematically enumerating all such merges for  $\mathcal{EL}++$ , we have implemented a grammar that expresses each aggregated pattern in English, relying on an intuitive choice of the best form of words: at a later stage we need to confirm that the resulting sentences are clearly understood, and to consider whether different formulations might be better.

## Acknowledgments

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/G033579/1 (SWAT: Semantic Web Authoring Tool). We thank our colleagues and the anonymous reviewers.



## References

- K. Bontcheva and Y. Wilks. 2004. Automatic report generation from ontologies: the MIAKT approach. In *Nineth International Conference on Applications of Natural Language to Information Systems (NLDB'2004)*, pages 214–225, Manchester, UK.
- Hercules Dalianis and Eduard H. Hovy. 1996. Aggregation in natural language generation. In *EWNLG '93: Selected papers from the Fourth European Workshop on Trends in Natural Language Generation, An Artificial Intelligence Perspective*, pages 88–105, London, UK. Springer-Verlag.
- Paolo Dongilli. 2008. Natural language rendering of a conjunctive query. Technical Report Knowledge Representation Meets Databases (KRDB) Research Centre Technical Report: KRDB08-3, Free University of Bozen-Bolzano.
- Dimitrios Galanis and Ion Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *Proceedings of the 11th European Workshop on Natural Language Generation*, pages 143–146, Morristown, NJ, USA. Association for Computational Linguistics.
- Dimitrios Galanis, George Karakatsiotis, Gerasimos Lampouras, and Ion Androutsopoulos. 2009. An open-source natural language generator for OWL ontologies and its use in protégé, and second life. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 17–20, Morristown, NJ, USA. Association for Computational Linguistics.
- Feikje Hielkema. 2009. *Using Natural Language Generation to Provide Access to Semantic Metadata*. Ph.D. thesis, University of Aberdeen.
- Kaarel Kaljurand and Norbert Fuchs. 2007. Verbalizing OWL in Attempto Controlled English. In *Proceedings of the Third International Workshop on OWL: Experiences and Directions OWLED 2007*.
- Holger Knublauch, Ray W. Ferguson, Natalya Fridman Noy, and Mark A. Musen. 2004. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *International Semantic Web Conference*, pages 229–243.
- Richard Power. 2010. Complexity assumptions in ontology verbalisation. In *48th Annual Meeting of the Association for Computational Linguistics*.
- Michael Reape and Chris Mellish. 1999. Just what is aggregation anyway? In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 20–29, Toulouse, France.
- Rolf Schwitter, Kaarel Kaljur, Anne Cregan, Catherine Dolbear, and Glen Hart. 2008. A comparison of three controlled natural languages for owl 1.1. In *4th OWL Experiences and Directions Workshop (OWLED 2008)*.
- Paul Smart. 2008. Controlled Natural Languages and the Semantic Web. Technical Report Technical Report ITA/P12/SemWebCNL, School of Electronics and Computer Science, University of Southampton).



# Paraphrase Generation as Monolingual Translation: Data and Evaluation

Sander Wubben, Antal van den Bosch, Emiel Krahmer

Tilburg centre for Cognition and Communication

Tilburg University

Tilburg, The Netherlands

{s.wubben, antal.vdnbosch, e.j.krahmer}@uvt.nl

## Abstract

In this paper we investigate the automatic generation and evaluation of sentential paraphrases. We describe a method for generating sentential paraphrases by using a large aligned monolingual corpus of news headlines acquired automatically from Google News and a standard Phrase-Based Machine Translation (PBMT) framework. The output of this system is compared to a word substitution baseline. Human judges prefer the PBMT paraphrasing system over the word substitution system. We demonstrate that BLEU correlates well with human judgements provided that the generated paraphrased sentence is sufficiently different from the source sentence.

## 1 Introduction

Text-to-text generation is an increasingly studied subfield in natural language processing. In contrast with the typical natural language generation paradigm of converting concepts to text, in text-to-text generation a source text is converted into a target text that approximates the meaning of the source text. Text-to-text generation extends to such varied tasks as summarization (Knight and Marcu, 2002), question-answering (Lin and Pantel, 2001), machine translation, and paraphrase generation.

Sentential paraphrase generation (SPG) is the process of transforming a source sentence into a target sentence in the same language which differs in form from the source sentence, but approximates its meaning. Paraphrasing is often used as a subtask in more complex NLP applications to allow for more variation in text strings presented as input, for example to generate paraphrases of questions that in their original form cannot be answered (Lin and Pantel, 2001; Riezler et al., 2007),

or to generate paraphrases of sentences that failed to translate (Callison-Burch et al., 2006). Paraphrasing has also been used in the evaluation of machine translation system output (Russo-Lassner et al., 2006; Kauchak and Barzilay, 2006; Zhou et al., 2006). Adding certain constraints to paraphrasing allows for additional useful applications. When a constraint is specified that a paraphrase should be shorter than the input text, paraphrasing can be used for sentence compression (Knight and Marcu, 2002; Barzilay and Lee, 2003) as well as for text simplification for question answering or subtitle generation (Daelemans et al., 2004).

We regard SPG as a monolingual machine translation task, where the source and target languages are the same (Quirk et al., 2004). However, there are two problems that have to be dealt with to make this approach work, namely obtaining a sufficient amount of examples, and a proper evaluation methodology. As Callison-Burch et al. (2008) argue, automatic evaluation of paraphrasing is problematic. The essence of SPG is to generate a sentence that is structurally different from the source. Automatic evaluation metrics in related fields such as machine translation operate on a notion of similarity, while paraphrasing centers around achieving dissimilarity. Besides the evaluation issue, another problem is that for an data-driven MT account of paraphrasing to work, a large collection of data is required. In this case, this would have to be pairs of sentences that are paraphrases of each other. So far, paraphrasing data sets of sufficient size have been mostly lacking. We argue that the headlines aggregated by Google News offer an attractive avenue.

## 2 Data Collection

Currently not many resources are available for paraphrasing; one example is the Microsoft Paraphrase Corpus (MSR) (Dolan et al., 2004; Nelken and Shieber, 2006), which with its 139,000 aligned

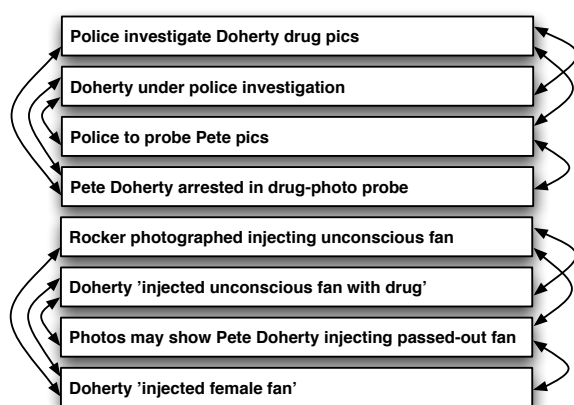


Figure 1: Part of a sample headline cluster, with aligned paraphrases

paraphrases can be considered relatively small. In this study we explore the use of a large, automatically acquired aligned paraphrase corpus. Our method consists of crawling the headlines aggregated and clustered by Google News and then aligning paraphrases within each of these clusters. An example of such a cluster is given in Figure 1. For each pair of headlines in a cluster, we calculate the Cosine similarity over the word vectors of the two headlines. If the similarity exceeds a defined upper threshold it is accepted; if it is below a defined lower threshold it is rejected. In the case that it lies between the thresholds, the process is repeated but then with word vectors taken from a snippet from the corresponding news article. This method, described in earlier work Wubben et al. (2009), was reported to yield a precision of 0.76 and a recall of 0.41 on clustering actual Dutch paraphrases in a headline corpus. We adapted this method to English. Our data consists of English headlines that appeared in Google News over the period of April to September 2006. Using this method we end up with a corpus of 7,400,144 pairwise alignments of 1,025,605 unique headlines<sup>1</sup>.

### 3 Paraphrasing methods

In our approach we use the collection of automatically obtained aligned headlines to train a paraphrase generation model using a Phrase-Based MT framework. We compare this approach to a word substitution baseline. The generated paraphrases along with their source head-

<sup>1</sup>This list of aligned pairs is available at <http://ilk.uvt.nl/~swubben/resources.html>

lines are presented to human judges, whose ratings are compared to the BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004) automatic evaluation metrics.

#### 3.1 Phrase-Based MT

We use the MOSES package to train a Phrase-Based Machine Translation model (PBMT) (Koehn et al., 2007). Such a model normally finds a best translation  $\tilde{e}$  of a text in language  $f$  to a text in language  $e$  by combining a translation model  $p(f|e)$  with a language model  $p(e)$ :

$$\tilde{e} = \arg \max_{e \in e^*} p(f|e)p(e)$$

GIZA++ is used to perform the word alignments (Och and Ney, 2003) which are then used in the Moses pipeline to generate phrase alignments in order to build the paraphrase model. We first tokenize our data before training a recaser. We then lowercase all data and use all unique headlines in the training data to train a language model with the SRILM toolkit (Stolcke, 2002). Then we invoke the GIZA++ aligner using the 7M training paraphrase pairs. We run GIZA++ with standard settings and we perform no optimization. Finally, we use the MOSES decoder to generate paraphrases for our test data.

Instead of assigning equal weights to language and translation model, we assign a larger weight of 0.7 to the language model to generate better formed (but more conservative) paraphrases. Because dissimilarity is a factor that is very important for paraphrasing but not implemented in a PBMT model, we perform post-hoc reranking of the different candidate outputs based on dissimilarity. For each headline in the testset we generate the ten best paraphrases as scored by the decoder and then rerank them according to dissimilarity to the source using the Levenshtein distance measure at the word level. The resulting headlines are re-cased using the previously trained recaser.

#### 3.2 Word Substitution

We compare the PBMT results with a simple word substitution baseline. For each noun, adjective and verb in the sentence this model takes that word and its Part of Speech tag and retrieves from WordNet its most frequent synonym from the most frequent synset containing the input word. We use the Memory Based Tagger (Daelemans et al., 1996)

System	Headline
Source	Florida executes notorious serial killer
PBMT	Serial killer executed in Florida
Word Sub.	Florida executes ill-famed series slayer
Source	Dublin evacuates airport due to bomb scare
PBMT	Dublin airport evacuated after bomb threat
Word Sub.	Dublin evacuates airdrome due to bomb panic
Source	N. Korea blasts nuclear sanctions
PBMT	N. Korea nuclear blast of sanctions
Word Sub.	N. Korea blasts atomic sanctions

Table 1: Examples of generated paraphrased headlines

trained on the Brown corpus to generate the POS-tags. The WordNet::QueryData<sup>2</sup> Perl module is used to query WordNet (Fellbaum, 1998). Generated headlines and their source for both systems are given in Table 1.

## 4 Evaluation

For the evaluation of the generated paraphrases we set up a human judgement study, and compare the human judges' ratings to automatic evaluation measures in order to gain more insight in the automatic evaluation of paraphrasing.

### 4.1 Method

We randomly select 160 headlines that meet the following criteria: the headline has to be comprehensible without reading the corresponding news article, both systems have to be able to produce a paraphrase for each headline, and there have to be a minimum of eight paraphrases for each headline. We need these paraphrases as multiple references for our automatic evaluation measures to account for the diversity in real-world paraphrases, as the aligned paraphrased headlines in Figure 1 witness.

The judges are presented with the 160 headlines, along with the paraphrases generated by both systems. The order of the headlines is randomized, and the order of the two paraphrases for each headline is also randomized to prevent a bias towards one of the paraphrases. The judges are asked to rate the paraphrases on a 1 to 7 scale, where 1 means that the paraphrase is very bad and 7 means that the paraphrase is very good. The judges were instructed to base their overall quality judgement on whether the meaning was retained, the paraphrase was grammatical and fluent, and whether the paraphrase was in fact different from

<sup>2</sup><http://search.cpan.org/dist/WordNet-QueryData/QueryData.pm>

system	mean	stdev.
PBMT	4.60	0.44
Word Substitution	3.59	0.64

Table 2: Results of human judgements ( $N = 10$ )

the source sentence. Ten judges rated two paraphrases per headline, resulting in a total of 3,200 scores. All judges were blind to the purpose of the evaluation and had no background in paraphrasing research.

### 4.2 Results

The average scores assigned by the human judges to the output of the two systems are displayed in Table 2. These results show that the judges rated the quality of the PBMT paraphrases significantly higher than those generated by the word substitution system ( $t(18) = 4.11, p < .001$ ).

Results from the automatic measures as well as the Levenshtein distance are listed in Table 3. We use a Levenshtein distance over tokens. First, we observe that both systems perform roughly the same amount of edit operations on a sentence, resulting in a Levenshtein distance over words of 2.76 for the PBMT system and 2.67 for the Word Substitution system. BLEU, METEOR and three typical ROUGE metrics<sup>3</sup> all rate the PBMT system higher than the Word Substitution system. Notice also that the all metrics assign the highest scores to the original sentences, as is to be expected: because every operation we perform is in the same language, the source sentence is also a paraphrase of the reference sentences that we use for scoring our generated headline. If we pick a random sentence from the reference set and score it against the rest of the set, we obtain similar scores. This means that this score can be regarded as an upper bound score for paraphrasing: we can not expect our paraphrases to be better than those produced by humans. However, this also shows that these measures cannot be used directly as an automatic evaluation method of paraphrasing, as they assign the highest score to the "paraphrase" in which nothing has changed. The scores observed in Table 3 do indicate that the paraphrases gener-

<sup>3</sup>ROUGE-1, ROUGE-2 and ROUGE-SU4 are also adopted for the DUC 2007 evaluation campaign, <http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html>

System	BLEU	ROUGE-1	ROUGE-2	ROUGE-SU4	METEOR	Lev.dist.	Lev. stdev.
PBMT	50.88	0.76	0.36	0.42	0.71	2.76	1.35
Wordsub.	24.80	0.59	0.22	0.26	0.54	2.67	1.50
Source	60.58	0.80	0.45	0.47	0.77	0	0

Table 3: Automatic evaluation and sentence Levenshtein scores

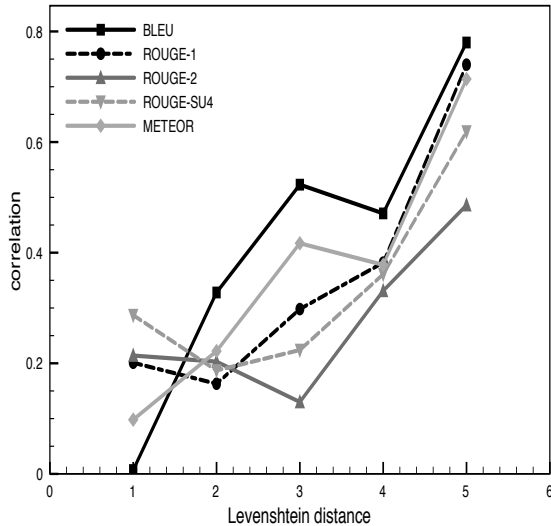


Figure 2: Correlations between human judgements and automatic evaluation metrics for various edit distances

ated by PBMT are less well formed than the original source sentence.

There is an overall medium correlation between the BLEU measure and human judgements ( $r = 0.41, p < 0.001$ ). We see a lower correlation between the various ROUGE scores and human judgements, with ROUGE-1 showing the highest correlation ( $r = 0.29, p < 0.001$ ). Between the two lies the METEOR correlation ( $r = 0.35, p < 0.001$ ). However, if we split the data according to Levenshtein distance, we observe that we generally get a higher correlation for all the tested metrics when the Levenshtein distance is higher, as visualized in Figure 2. At Levenshtein distance 5, the BLEU score achieves a correlation of 0.78 with human judgements, while ROUGE-1 manages to achieve a 0.74 correlation. Beyond edit distance 5, data sparsity occurs.

## 5 Discussion

In this paper we have shown that with an automatically obtained parallel monolingual corpus with several millions of paired examples, it is possible to develop an SPG system based on a PBMT

framework. Human judges preferred the output of our PBMT system over the output of a word substitution system. We have also addressed the problem of automatic paraphrase evaluation. We measured BLEU, METEOR and ROUGE scores, and observed that these automatic scores correlate with human judgements to some degree, but that the correlation is highly dependent on edit distance. At low edit distances automatic metrics fail to properly assess the quality of paraphrases, whereas at edit distance 5 the correlation of BLEU with human judgements is 0.78, indicating that at higher edit distances these automatic measures can be utilized to rate the quality of the generated paraphrases. From edit distance 2, BLEU correlates best with human judgements, indicating that MT evaluation metrics might be best for SPG evaluation.

The data we used for paraphrasing consists of headlines. Paraphrase patterns we learn are those used in headlines and therefore different from standard language. The advantage of our approach is that it paraphrases those parts of sentences that it can paraphrase, and leaves the unknown parts intact. It is straightforward to train a language model on in-domain text and use the translation model acquired from the headlines to generate paraphrases for other domains. We are also interested in capturing paraphrase patterns from other domains, but acquiring parallel corpora for these domains is not trivial.

Instead of post-hoc dissimilarity reranking of the candidate paraphrase sentences we intend to develop a proper paraphrasing model that takes dissimilarity into account in the decoding process. In addition, we plan to investigate if our paraphrase generation approach is applicable to sentence compression and simplification. On the topic of automatic evaluation, we aim to define an automatic paraphrase generation assessment score. A paraphrase evaluation measure should be able to recognize that a good paraphrase is a well-formed sentence in the source language, yet it is clearly dissimilar to the source.

## References

- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, June.
- Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 16–23.
- Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24.
- Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. 2008. Parametric: an automatic evaluation metric for paraphrasing. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 97–104.
- Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27.
- Walter Daelemans, Anja Hothker, and Erik Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in dutch and english. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1045–1048.
- Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 350.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*, May.
- David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462, June.
- Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris C. Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.
- Dekang Lin and Patrick Pantel. 2001. Dirt: Discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.
- Rani Nelken and Stuart M. Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, 3–7 April.
- Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.
- Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 142–149, July.
- Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *ACL*.
- Grazia Russo-Lassner, Jimmy Lin, and Philip Resnik. 2006. A paraphrase-based approach to machine translation evaluation. Technical report, University of Maryland, College Park.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *In Proc. Int. Conf. on Spoken Language Processing*, pages 901–904.
- Sander Wubben, Antal van den Bosch, Emiel Kraemer, and Erwin Marsi. 2009. Clustering and matching headlines for automatic paraphrase acquisition. In *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation*, pages 122–125.
- Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 77–84, July.





# Anchor-Progression in Spatially Situated Discourse: a Production Experiment

Hendrik Zender and Christopher Koppermann and Fai Greeve and Geert-Jan M. Kruijff

Language Technology Lab  
German Research Center for Artificial Intelligence (DFKI)  
Saarbrücken, Germany  
zender@dfki.de

## Abstract

The paper presents two models for producing and understanding situationally appropriate referring expressions (REs) during a discourse about large-scale space. The models are evaluated against an empirical production experiment.

## 1 Introduction and Background

For situated interaction, an intelligent system needs methods for relating entities in the world, its representation of the world, and the natural language references exchanged with its user. Human natural language processing and algorithmic approaches alike have been extensively studied for application domains restricted to small visual scenes and other small-scale surroundings. Still, rather little research has addressed the specific issues involved in establishing reference to entities outside the currently visible scene. The challenge that we address here is how the focus of attention can shift over the course of a discourse if the domain is larger than the currently visible scene.

The generation of referring expressions (GRE) has been viewed as an isolated problem, focussing on efficient algorithms for determining which information from the domain must be incorporated in a noun phrase (NP) such that this NP allows the hearer to optimally understand which referent is meant. The domains of such approaches usually consist of small, static domains or simple visual scenes. In their seminal work Dale and Reiter (1995) present the Incremental Algorithm (IA) for GRE. Recent extensions address some of its shortcomings, such as negated and disjointed properties (van Deemter, 2002) and an account of salience for generating contextually appropriate shorter REs (Krahmer and Theune, 2002). Other, alternative GRE algorithms exist (Horacek, 1997; Bateman, 1999; Krahmer et al., 2003). However, all these al-

gorithms rely on a given *domain of discourse* constituting the current *context* (or *focus of attention*). The task of the GRE algorithm is then to single out the intended referent against the other members of the context, which act as *potential distractors*. As long as the domains are such closed-context scenarios, the intended referent is always in the current focus. We address the challenge of producing and understanding of references to entities that are outside the current focus of attention, because they have not been mentioned yet and are beyond the currently observable scene.

Our approach relies on the dichotomy between *small-scale space* and *large-scale space* for human spatial cognition. Large-scale space is “a space which cannot be perceived at once; its global structure must be derived from local observations over time” (Kuipers, 1977). In everyday situations, an office environment, one’s house, or a university campus are large-scale spaces. A table-top or a part of an office are examples of small-scale space. Despite large-scale space being not fully observable, people can nevertheless have a reasonably complete mental representation of, e.g., their domestic or work environments in their *cognitive maps*. Details might be missing, and people might be uncertain about particular things and states of affairs that are known to change frequently. Still, people regularly engage in a conversation about such an environment, making successful references to spatially located entities.

It is generally assumed that humans adopt a *partially hierarchical* representation of spatial organization (Stevens and Coupe, 1978; McNamara, 1986). The basic units of such a representation are *topological* regions (i.e., more or less clearly bounded spatial areas) (Hirtle and Jonides, 1985). Paraboni et al. (2007) are among the few to address the issue of generating references to entities outside the immediate environment, and present an algorithm for *context determination* in hierar-

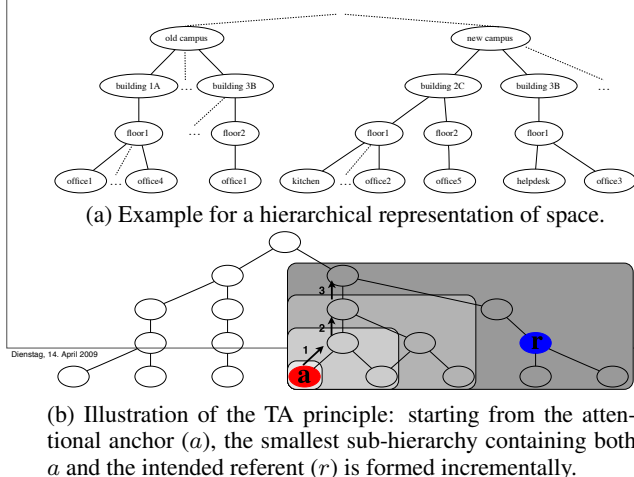


Figure 1: TA in a spatial hierarchy.

chically ordered domains. However, since it is mainly targeted at producing textual references to entities in written documents (e.g., figures and tables in book chapters), they do not address the challenges of physical and perceptual situatedness. Large-scale space can be viewed as a hierarchically ordered domain. To keep track of the referential context in such a domain, in our previous work we propose the principle of *topological abstraction* (TA, summarized in Fig. 1) for context extension (Zender et al., 2009a), similar to Ancestral Search (Paraboni et al., 2007). In (Zender et al., 2009b), we describe the integration of the approach in an NLP system for situated human-robot dialogues and present two algorithms instantiating the TA principle for GRE and resolving referring expressions (RRE), respectively. It relies on two parameters: the location of the *intended referent*  $r$ , and the *attentional anchor*  $a$ . As discussed in our previous works, for single utterances the anchor is the physical position where it is made (i.e., the *utterance situation* (Devlin, 2006)). Below, we propose models for attentional anchor-progression for longer discourses about large-scale space, and evaluate them against real-world data.

## 2 The Models

In order to account for the determination of the attentional anchor  $a$ , we propose a model called *anchor-progression*  $A$ . The model assumes that each *exophoric* reference<sup>1</sup> serves as *attentional anchor* for the subsequent reference. It is based on observations on “principles for anchoring resource situations” by Poesio (1993), where the expression of movement in the domain determines

<sup>1</sup>This excludes pronouns as well as other descriptions that pick up an existing referent from the linguistic context.

the updated current mutual focus of attention.  $a$  and  $r$  are then passed to the TA algorithm. Taking into account the verbal behavior observed in our experiment, we also propose a refined model of *anchor-resetting*  $R$ , where for each new turn (e.g., a new instruction), the anchor is re-set to the *utterance situation*.  $R$  leads to the inclusion of navigational information for each first RE in a turn, thus reassuring the hearer of the focus of attention.

## 3 The Experiment

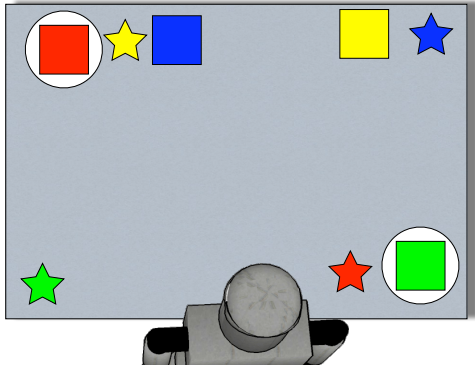
We are interested in the way the disambiguation strategies change when producing REs during a discourse about large-scale space versus discourse about small-scale space. In our experiment, we gathered a corpus of spoken instructions in two different situations: *small-scale space* (SSS) and *large-scale space* (LSS). We use the data to evaluate the utility of the  $A$  and  $R$  models. We specifically evaluate them against the traditional (*global*) model  $G$  in which the indented referent must be singled out from all entities in the domain.

The cover story for the experiment was to record spoken instructions to help improve a speech recognition system for robots. The participants were asked to imagine an intelligent service robot capable of understanding natural language and familiar with its environment. The task of the participants was to instruct the robot to clean up a working space, i.e., a table-top (SSS) and an indoor environment (LSS) by placing target objects (cookies or balls) in boxes of the same color. The use of color terms to identify objects was discouraged by telling the participants that the robot is unable to perceive color. The stimuli consisted of 8 corresponding scenes of the table-top and the domestic setting (cf. Fig. 2). In order to preclude the specific phenomena of collaborative, task-oriented dialogue (cf., e.g., (Garrod and Pickering, 2004)), the participants had to instruct an imaginary recipient of orders. The choice of a robot was made to rule out potential social implications when imagining, e.g., talking to a child, a butler, or a friend.

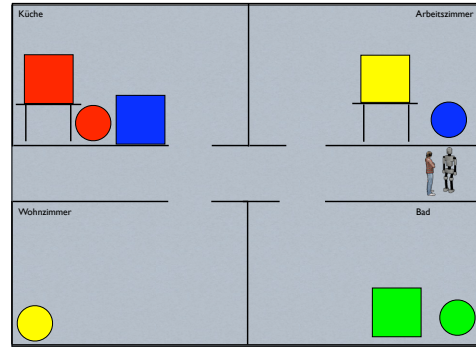
The SSS scenes show a bird’s-eye view of the table including the robot’s position (similar to (Funakoshi et al., 2004)). The way the objects are arranged allows to refer to their location with respect to the corners of the table, with plates as additional landmarks. The LSS scenes depict an indoor environment with a corridor and, parallel to SSS, four rooms with tables as landmarks. The scenes show

Table 1: Example from the small-scale (1–2) and large-scale space (3–4) scenes in Fig. 2.

1. *nimm [das plätzchen unten links]<sub>m<sub>G,A</sub></sub>*, *leg es [in die schachtel unten rechts auf dem teller]<sub>o<sub>G,A</sub></sub>*  
‘take the cookie on the bottom left, put it into the bottom right box on the plate’
2. *nimm [das plätzchen unten rechts]<sub>m<sub>G,o<sub>A</sub></sub></sub>*, *leg es [in die schachtel oben links auf dem teller]<sub>m<sub>G,A</sub></sub>*  
‘take the cookie on the bottom right, put it into the top left box on the plate’
3. *geh [ins wohnzimmer]<sub>m<sub>G,A,R</sub></sub> und nimm [den ball]<sub>u<sub>G,m<sub>A,R</sub></sub> und bring ihn [ins arbeitszimmer]<sub>m<sub>G,A,R</sub></sub>, leg ihn [in die kiste auf dem tisch]<sub>u<sub>G,o<sub>A,R</sub></sub></sub></sub>*  
‘go to the living room and take the ball and bring it to the study; put it into the box on the table’
4. *und nimm [den ball]<sub>u<sub>G,R,m<sub>A</sub></sub> und bring ihn [in die küche]<sub>m<sub>G,A,R</sub></sub> und leg ihn [in die kiste auf dem boden]<sub>u<sub>G,m<sub>A,R</sub></sub></sub></sub>*  
‘and take the ball and bring it to the kitchen and put it into the box on the floor’



(a) Small-scale space: squares represent small boxes, stars cookies, and white circles plates.



(b) Large-scale space: squares represent boxes placed on the floor or on a table, circles represent balls, rooms are labeled.

Figure 2: Two stimuli scenes from the experiment.

the robot and the participant in the corridor.

In order to gather more comparable data we opted for a *within-participants* approach. Each person participated in the *SSS treatment* and in the *LSS treatment*. To counterbalance potential carry-over effects, half of the participants were shown the treatments in inverse order, and the sequence of the 8 scenes in each treatment was varied in a principled way. In order to make the participants produce multi-utterance discourses, they were required to refer to all target object pairs. The exact wording of their instructions was up to them.

Participants were placed in front of a screen and a microphone into which they spoke their orders to the imaginary robot, followed by a self-paced keyword after which the experimenter showed the next scene. The experiment was conducted in German and consisted of a pilot study (10 participants) and the main part (19 female and 14 male students, aged 19–53, German native speakers). The data of three participants who did not behave according to the instructions was discarded. The individual sessions took 20–35 min., and the participants were paid for their efforts.

Using the UAM CorpusTool software, transcriptions of the recorded spoken instructions were annotated for occurrences of the linguistic phenomenon we are interested in, i.e., REs. Sam-

ples were cross-checked by a second annotator. REs were marked as shallow ‘refex’ segments, i.e., complex NPs were not decomposed into their constituents. Only definite NPs representing exophoric REs (cf. Sec. 2) qualify as ‘refex’ segments. If a turn contained an indefinite NP, the whole turn was discarded. The ‘refex’ segments were coded according to the amount of information they contain, and under which disambiguation model  $M \in \{G, A, R\}$  ( $R$  only for LSS) they succeed in singling out the described referent. Following Engelhardt et al. (2006), we distinguish three types of semantic specificity. A RE is an *over-description* with respect to  $M$  ( $over_M$ ) if it contains redundant information, and it is an *under-description* ( $under_M$ ) if it is ambiguous according to  $M$ . *Minimal descriptions* ( $min_M$ ) contain just enough information to uniquely identify the referent. Table 1 shows annotated examples.

## 4 Results

The collected corpus consists of 30 annotated sessions with 2 treatments comprising 8 scenes with 4 turns. In total, it contains 4,589 annotated REs, out of which only 83 are errors. Except for the error rate calculation, we only consider non-error ‘refex’ segments as the universe. The SSS treat-

Table 2: Mean frequencies (with standard deviation in italics) of minimal (*min*), over-descriptions (*over*), and under-descriptions (*under*) with respect to the models (*A*, *R*, *G*) in both treatments.

	<i>over<sub>G</sub></i>	<i>over<sub>A</sub></i>	<i>over<sub>R</sub></i>	<i>min<sub>G</sub></i>	<i>min<sub>A</sub></i>	<i>min<sub>R</sub></i>	<i>under<sub>G</sub></i>	<i>under<sub>A</sub></i>	<i>under<sub>R</sub></i>
small-scale space	13.94% <i>15.85%</i>	34.45% <i>14.37%</i>		78.90% <i>17.66%</i>	60.11% <i>13.13%</i>		7.16% <i>12.07%</i>	5.43% <i>10.50%</i>	
large-scale space	6.81% <i>7.53%</i>	34.75% <i>12.13%</i>	20.06 % <i>10.10%</i>	68.04% <i>17.87%</i>	64.55% <i>13.13%</i>	76.73% <i>10.66%</i>	25.16% <i>19.48%</i>	0.69% <i>1.72%</i>	3.21% <i>5.06%</i>

ment contains 1,902 ‘refex’, with a mean number of 63.4 and a std. dev.  $\sigma=1.98$  per participant. This corresponds to the expected number of 64 REs to be uttered: 8 scenes  $\times$  4 target object pairs. The LSS treatment contains 2,604 ‘refex’ with an average of 86.8 correct REs ( $\sigma=18.19$ ) per participant. As can be seen in Table 1 (3–4), this difference is due to the participants’ referring to intermediate waypoints in addition to the target objects. Table 2 summarizes the analysis of the annotated data.

Overall, the participants had no difficulties with the experiment. The mean error rates are low in both treatments: 1.78% ( $\sigma=3.36\%$ ) in SSS, and 1.80% ( $\sigma=2.98\%$ ) in LSS. A paired sample t-test of both scores for each participant shows that there is no significant difference between the error rates in the treatments ( $p=0.985$ ), supporting the claim that both treatments were of equal difficulty. Moreover, a MANOVA shows no significant effect of treatment-order for the verbal behavior under study, ruling out potential carry-over effects.

Production experiments always exhibit a considerable variation between participants. When modeling natural language processing systems, one needs to take this into account. A GRE component should produce REs that are easy to understand, i.e., ambiguities should be avoided and over-descriptions should occur sparingly. A GRE algorithm will always try to produce minimal descriptions. The generation of an under-description means a failure to construct an identifying RE, while over-descriptions are usually the result of a globally ‘bad’ incremental construction of the generated REs (as is the case, e.g., in the IA). An RRE component, on the other hand, should be able to identify as many referents as possible by treating as few as possible REs as under-descriptions.

The analysis of the SSS data with respect to *G* establishes the baseline for a comparison with other experiments and GRE approaches. 13.9% of the REs contain redundant information (*over<sub>G</sub>*), compared to 21% in (Viethen and Dale, 2006). In contrast, however, our SSS scenes did not provide the possibility for producing more-than-minimal REs for every target object, which might account

for the difference. *under<sub>G</sub>* REs occur with a frequency of 7.2% in the SSS data. Because under-descriptions result in the the hearer being unable to reliably resolve the reference, this means that the robot in our experiment cannot fulfill its task. This might explain the difference to the 16% observed in the task-independent study by Viethen and Dale (2006). The significantly ( $p<0.001$ ) higher mean frequency of *min<sub>G</sub>* than *min<sub>A</sub>* underpins that *G* is an accurate model for the verbal behavior in SSS. However, *G* does not fit the LSS data well. An RRE algorithm with model *G* would fail to resolve the intended referent in 1 out of 4 cases (cf. *under<sub>G</sub>* in LSS). With only 0.7% *under<sub>A</sub>* REs on average, *A* models the LSS data significantly better ( $p<0.001$ ). Still, there is a high rate of *over<sub>A</sub>* REs. In comparison, *R* yields a significantly ( $p<0.001$ ) lower amount of *over<sub>R</sub>*. The mean frequency of *under<sub>R</sub>* is significantly ( $p=0.010$ ) higher than for *under<sub>A</sub>*, but still below *under<sub>G</sub>* in the SSS data. With a mean frequency of 76.7% *min<sub>R</sub>*, *R* models the data better than both *G* and *A*. For the REs in LSS *min<sub>R</sub>* is in the same range as *min<sub>G</sub>* for the REs in SSS.

## 5 Conclusions

Overall, the data exhibit a high mean frequency of over-descriptions. However, since this means that the human-produced REs contain more information than minimally necessary, this does not negatively affect the performance of an RRE algorithm. For a GRE algorithm, however, a more cautious approach might be desirable. In situated discourse about LSS, we thus suggest that *A* is suitable for the RRE task because it yields the least amount of unresolvable under-descriptions. For the GRE task *R* is more appropriate. It strikes a balance between producing short descriptions and supplementing navigational information.

## Acknowledgments

This work was supported by the EU Project CogX (FP7-ICT-215181). Thanks to Mick O’Donnell for his support with the UAM CorpusTool.

## References

- John A. Bateman. 1999. Using aggregation for selecting content when generating referring expressions. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL'99)*, pages 127–134, Morristown, NJ, USA.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Keith Devlin. 2006. Situation theory and situation semantics. In Dov M. Gabbay and John Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 601–664. Elsevier.
- Paul E. Engelhardt, Karl G.D. Bailey, and Fernanda Ferreira. 2006. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, 54(4):554–573.
- Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. 2004. Generation of relative referring expressions based on perceptual grouping. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA.
- Simon Garrod and Martin J. Pickering. 2004. Why is conversation so easy? *Trends in Cognitive Sciences*, 8(1):8–11, January.
- Stephen C. Hirtle and John Jonides. 1985. Evidence for hierarchies in cognitive maps. *Memory and Cognition*, 13:208–217.
- Helmut Horacek. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-97)*, pages 206–213, Morristown, NJ, USA.
- Emiel Krahmer and Mariët Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and R. Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford, CA, USA.
- Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- Benjamin Kuipers. 1977. *Representing Knowledge of Large-scale Space*. PhD thesis, MIT-AI TR-418, Massachusetts Institute of Technology, Cambridge, MA, USA, May.
- Timothy P. McNamara. 1986. Mental representations of spatial relations. *Cognitive Psychology*, 18:87–121.
- Ivandr  Paraboni, Kees van Deemter, and Judith Masthoff. 2007. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254, June.
- Massimo Poesio. 1993. A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues. In Peter Aczel, David Israel, Yasuhiro Katagiri, and Stanley Peters, editors, *Situation Theory and its Applications Volume 3*, CSLI Lecture Notes No. 37, pages 339–374. Center for the Study of Language and Information, Menlo Park, CA, USA.
- Albert Stevens and Patty Coupe. 1978. Distortions in judged spatial relations. *Cognitive Psychology*, 10:422–437.
- Kees van Deemter. 2002. Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52.
- Jette Viethen and Robert Dale. 2006. Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 63–70, Sydney, Australia.
- Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayova. 2009a. A situated context model for resolution and generation of referring expressions. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 126–129, Athens, Greece, March.
- Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayova. 2009b. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1604–1609, Pasadena, CA, USA, July.



# Generation Challenges 2010





## Preface

Generation Challenges 2010 was the fourth round of shared-task evaluation competitions (STECs) that involve the generation of natural language; it followed the Pilot Attribute Selection for Generating Referring Expressions Challenge in 2007 (AS-GRE'07) and Referring Expression Generation Challenges in 2008 (REG'08), and Generation Challenges 2009 (GenChal'09). More information about all these NLG STEC activities can be found via the links on the Generation Challenges homepage (<http://www.nltg.brighton.ac.uk/research/genchal10>).

Generation Challenges 2010 brought together three sets of STECs: the three GREC Challenges, GREC Named Entity Generation (GREC-NEG), Named Entity Reference Detection (GREC-NER), and Named Entity Reference Regeneration (GREC-Full), organised by Anja Belz and Eric Kow; the Challenge on Generating Instructions in Virtual Environments (GIVE) organised by Donna Byron, Justine Cassell, Robert Dale, Alexander Koller, Johanna Moore, Jon Oberlander, and Kristina Striegnitz; and the new Question Generation (QG) tasks, organised by Vasile Rus, Brendan Wyse, Mihai Lintean, Svetlana Stoyanchev and Paul Piwek.

In the GIVE Challenge, participating teams developed systems which generate natural-language instructions to users navigating a virtual 3D environment and performing computer-game-like tasks. The seven participating systems were evaluated by measuring how quickly, accurately and efficiently users were able to perform tasks with a given system's instructions, as well as on subjective measures. Unlike the first GIVE Challenge, this year's challenge allowed users to move and turn freely in the virtual environment, rather than in discrete steps, making the NLG task much harder. The evaluation report for the GIVE Challenge can be found in this volume; the participants' reports will be made available on the GIVE website (<http://www.give-challenge.org/research>) at a later stage.

The GREC Tasks used the GREC-People corpus of introductory sections from Wikipedia articles on people. In GREC-NEG, the task was to select referring expressions for all mentions of all people in an article from given lists of alternatives (this was the same task as at GenChal'09). The GREC-NER task combines named-entity recognition and coreference resolution, restricted to people entities; the aim for participating systems is to identify all those types of mentions of people that are annotated in the GREC-People corpus. The aim for GREC-Full systems was to improve the referential clarity and fluency of input texts. Participants were free to do this in whichever way they chose. Participants were encouraged, though not required, to create systems which replace referring expressions as and where necessary to produce as clear and fluent a text as possible. This task could be viewed as combining the GREC-NER and GREC-NEG tasks.

The first Question Generation challenge consisted of three tasks: Task A required questions to be generated from paragraphs of texts; Task B required systems to generate questions from sentences, and Task C was an Open Task track in which any QG research involving evaluation could be submitted. At the time of going to press, the QG tasks are still running; this volume contains a preliminary report from the organisers.

In addition to the four shared tasks, Generation Challenges 2010 offered (i) an open submission track in which participants could submit any work involving the data from any of the shared tasks, while opting out of the competitive element, (ii)

an evaluation track, in which proposals for new evaluation methods for the shared task could be submitted, and (iii) a task proposal track in which proposals for new shared tasks could be submitted. We believe that these types of open-access tracks are important because they allow the wider research community to shape the focus and methodologies of STECs directly.

We received three submissions in the Task Proposals track: an outline proposal for tasks involving language generation under uncertainty (Lemon et al.); a proposal for a shared task on improving text written by non-native speakers (Dale and Kilgarriff); and a proposal for a surface realisation task (White et al.).

Once again, we successfully applied (with the help of support letters from many of last year's participants and other HLT colleagues) for funding from the Engineering and Physical Sciences Research Council (EPSRC), the main funding body for HLT in the UK. This support helped with all aspects of organising Generation Challenges 2010, and enabled us to create the new GREC-People corpus and to carry out extensive human evaluations, as well as to employ a dedicated research fellow (Eric Kow) to help with all aspects of Generation Challenges 2010.

Preparations are already underway for a fifth NLG shared-task evaluation event next year, Generation Challenges 2011, which is likely to include a further run of the GIVE Task, a second run of the QG Challenge, and a pilot surface realisation task. We expect that results will be presented at ENLG'11.

Just like our previous STECs, Generation Challenges 2010 would not have been possible without the contributions of many different people. We would like to thank the students of Oxford University, KCL, UCL, Brighton and Sussex Universities who participated in the evaluation experiments, as well as all other participants in our online data elicitation and evaluation exercises; the INLG'10 organisers, Ielka van der Sluis, John Kelleher and Brian MacNamee; the research support team at Brighton University and the EPSRC for help with obtaining funding; and last but not least, the participants in the shared tasks themselves.

*July 2010*

*Anja Belz, Albert Gatt and Alexander Koller*

# The GREC Challenges 2010: Overview and Evaluation Results

Anja Belz      Eric Kow

Natural Language Technology Group  
School of Computing, Mathematical and Information Sciences  
University of Brighton  
Brighton BN2 4GJ, UK  
{asb,eykk10}@bton.ac.uk

## Abstract

There were three GREC Tasks at Generation Challenges 2010: GREC-NER required participating systems to identify all people references in texts; for GREC-NEG, systems selected coreference chains for all people entities in texts; and GREC-Full combined the NER and NEG tasks, i.e. systems identified and, if appropriate, replaced references to people in texts. Five teams submitted 10 systems in total, and we additionally created baseline systems for each task. Systems were evaluated automatically using a range of intrinsic metrics. In addition, systems were assessed by human judges using preference strength judgements. This report presents the evaluation results, along with descriptions of the three GREC tasks, the evaluation methods, and the participating systems.

## 1 Introduction

Until recently, referring expression generation (REG) research focused on the task of selecting the semantic content of one-off mentions of listener-familiar discourse entities. In the GREC research programme we have been interested in REG as (i) grounded within discourse context, (ii) embedded within an application context, and (iii) informed by naturally occurring data.

In general terms, the GREC tasks are about how to select appropriate references to an entity in the context of a piece of discourse longer than a sentence. In GREC'10, there were three subtasks: identification of references to people in free text (GREC-NER); selection of references to people in text (GREC-NEG); and regeneration of references to people in text (GREC-Full) which can be thought of as combining the NER and NEG tasks.

The immediate motivating application context

for the GREC Tasks is the improvement of referential clarity and coherence in extractive summaries and multiply edited texts (such as Wikipedia articles) by regenerating referring expressions contained in them. The motivating theoretical interest for the GREC Tasks is to discover what kind of information is useful for making choices between different kinds of referring expressions in context.

The GREC'10 tasks used the GREC-People corpus which consists of 1,100 Wikipedia texts about people within which we have annotated all references to people.

Five teams participated in the GREC'10 tasks (see Table 1), submitting 10 systems in total. Two of these were created by combining the NER system of one of the teams with the NEG systems of two different teams, producing two 'combined' systems for the Full Task. We also used the corpus texts themselves as 'system' outputs, and created baseline systems for all three tasks. We evaluated systems using a range of intrinsic automatically computed and human-assessed evaluation methods. This report describes the data (Section 2) and evaluation methods (Section 3) used in the three GREC'10 tasks, and then presents task definition, participating systems, evaluation methods, and evaluation results for each of the three tasks separately (Sections 4–6).

## 2 GREC'10 Data

The GREC'10 data is derived from the GREC-People corpus which (in its 2010 version) consists of 1,100 annotated introduction sections from Wikipedia articles in the category People. An introduction section was defined as the textual content of a Wikipedia article from the title up to (and excluding) the first section heading, the table of contents or the end of the text, whichever comes first. Each text belongs to one of six subcategories: inventors, chefs, early music composers, explorers, kickboxers and romantic composers. For the

Team	Affiliation	NEG systems	NER systems	Full systems
UDel <sup>x</sup>	University of Delaware	UDel-NEG	UDel-NER	UDel-Full
UMUS	Université du Maine Universität Stuttgart	UMUS	–	–
JU <sup>x</sup>	Jadavpur University	JU	–	–
Poly-co	École Polytechnique de Montréal	–	Poly-co	–
XRCE <sup>y</sup>	Xerox Research Centre Europe	XRCE	–	–
UDel/UMUS	(see above)	–	–	UDel-UMUS-Full
UDel/XRCE	(see above)	–	–	UDel-XRCE-Full

Table 1: GREC-NEG’09 teams and systems (combined teams in last two rows). <sup>x</sup> = resubmitted after fixing character encoding problems and/or software bugs; <sup>y</sup> = late submission.

	All	Inventors	Chefs	Early Composers	Explorers	Kickboxers	Romantic Composers
Training	809	249	248	312	–	–	–
Development	91	28	28	35	–	–	–
Test (NEG)	100	31	30	39	–	–	–
Test (NER/Full)	100	–	–	–	33	34	33
Total	1,100	307	306	387	33	34	33

Table 2: Overview of GREC’10 data sets.

purposes of the GREC task, the GREC-People corpus was divided into training, development and test data. The number of texts in the subsets are as shown in Table 2.

In the GREC-People annotation scheme, a distinction is made between *reference* and *referential expression*. A reference is ‘an instance of referring’ which is unique, whereas a referential expression is a word string and each reference can be realised by many different referential expressions. In the GREC corpora, each time an entity is referred to, there is a single reference, but there may be one or several referring expressions provided with it: in the training/development data, there is a single RE for each reference (the one found in the corpus); in the test set, there are four REs for each reference (the one from the corpus and three additional ones selected by subjects in a manual selection experiment).

We first manually annotated people mentions in the GREC-People texts by marking up the word strings that function as referential expressions (REs) and annotating them with coreference information as well as semantic category, syntactic category and function, and various supplements and dependents. Annotations included nested references, plurals and coordinated REs, certain unnamed references and indefinites. In terminology and the treatment of syntax used in the annotation scheme we relied heavily on *The Cambridge Grammar of the English Language* by Huddleston and Pullum (2002). For full details of the manual

annotation please refer to the GREC’10 documentation (Belz, 2010).

The manual annotations were then automatically checked and converted to XML format. In the XML format of the annotations, the beginning and end of a reference is indicated by `<REF><REFEX> . . . </REFEX></REF>` tags, and other properties mentioned above (e.g. syntactic category) are encoded as attributes on these tags. For the GREC tasks we decided not to transfer the annotations of integrated dependents and relative clauses to the XML format. Such dependents are included within `<REFEX> . . . </REFEX>` annotations where appropriate, but without being marked up as separate constituents.

Figure 1 shows one of the XML-annotated texts from the GREC data. For full details of the manual annotations and the XML version, please refer to the GREC’10 documentation (Belz, 2010). Here we provide a brief summary.

The `REF` element indicates a reference, and is composed of one `REFEX` element (the ‘selected’ referential expression for the given reference; in the corpus texts it is the referential expression found in the corpus). The attributes of the `REF` element are `ENTITY` (entity identifier), `MENTION` (mention identifier), `SEMLOC` (semantic category), `SYNLOC` (syntactic category), and `SYNFUNC` (syntactic function). `ENTITY` and `MENTION` together constitute a unique identifier for a reference within a text; together with the `TEXT ID`, they constitute a unique identifier for a reference within the entire

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE GREC-ITEM SYSTEM "genchal09-grec.dtd">
<GREC-ITEM>
<TEXT ID="15">
<TITLE>Alexander Fleming</TITLE>

<PARAGRAPH> <REF ENTITY="0" MENTION="1" SEMCAT="person" SYNCAT="np" SYNFUNC="subj">
  <REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Sir Alexander Fleming</REFEX>
</REF> (6 August 1881 - 11 March 1955) was a Scottish biologist and pharmacologist.
<REF ENTITY="0" MENTION="2" SEMCAT="person" SYNCAT="np" SYNFUNC="subj">
  <REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Fleming</REFEX>
</REF> published many articles on bacteriology, immunology, and chemotherapy.
<REF ENTITY="0" MENTION="3" SEMCAT="person" SYNCAT="np" SYNFUNC="subj-det">
  <REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="genitive">his</REFEX>
</REF> best-known achievements are the discovery of the enzyme lysozyme in 1922 and the discovery
of the antibiotic substance penicillin from the fungus Penicillium notatum in 1928, for which
<REF ENTITY="0" MENTION="4" SEMCAT="person" SYNCAT="np" SYNFUNC="subj">
  <REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="nominative">he</REFEX>
</REF> shared the Nobel Prize in Physiology or Medicine in 1945 with
<REF ENTITY="1" MENTION="1" SEMCAT="person" SYNCAT="np" SYNFUNC="obj">
  <REFEX ENTITY="1" REG08-TYPE="name" CASE="plain">Florey</REFEX>
  and
<REF ENTITY="2" MENTION="1" SEMCAT="person" SYNCAT="np" SYNFUNC="obj">
  <REFEX ENTITY="2" REG08-TYPE="name" CASE="plain">Chain</REFEX>
</REF>. </PARAGRAPH>
</TEXT>

<ALT-REFEX>
<REFEX ENTITY="0" REG08-TYPE="empty" CASE="no_case">_</REFEX>
<REFEX ENTITY="0" REG08-TYPE="name" CASE="genitive">Fleming's</REFEX>
<REFEX ENTITY="0" REG08-TYPE="name" CASE="genitive">Sir Alexander Fleming's</REFEX>
<REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Fleming</REFEX>
<REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Sir Alexander Fleming</REFEX>
<REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="accusative">him</REFEX>
<REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="genitive">his</REFEX>
<REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="nominative">he</REFEX>
<REFEX ENTITY="0" REG08-TYPE="pronoun" CASE="nominative">who</REFEX>
<REFEX ENTITY="1" REG08-TYPE="empty" CASE="no_case">_</REFEX>
<REFEX ENTITY="1" REG08-TYPE="name" CASE="genitive">Florey's</REFEX>
<REFEX ENTITY="1" REG08-TYPE="name" CASE="plain">Florey</REFEX>
<REFEX ENTITY="1" REG08-TYPE="pronoun" CASE="accusative">him</REFEX>
<REFEX ENTITY="1" REG08-TYPE="pronoun" CASE="genitive">his</REFEX>
<REFEX ENTITY="1" REG08-TYPE="pronoun" CASE="nominative">he</REFEX>
<REFEX ENTITY="1" REG08-TYPE="pronoun" CASE="nominative">who</REFEX>
<REFEX ENTITY="2" REG08-TYPE="empty" CASE="no_case">_</REFEX>
<REFEX ENTITY="2" REG08-TYPE="name" CASE="genitive">Chain's</REFEX>
<REFEX ENTITY="2" REG08-TYPE="name" CASE="plain">Chain</REFEX>
<REFEX ENTITY="2" REG08-TYPE="pronoun" CASE="accusative">him</REFEX>
<REFEX ENTITY="2" REG08-TYPE="pronoun" CASE="genitive">his</REFEX>
<REFEX ENTITY="2" REG08-TYPE="pronoun" CASE="nominative">he</REFEX>
<REFEX ENTITY="2" REG08-TYPE="pronoun" CASE="nominative">who</REFEX>
</ALT-REFEX>
</GREC-ITEM>

```

Figure 1: Example XML-annotated text from the GREC-NEG'09 data.

corpus.

A REFEX element indicates a referential expression (a word string that can be used to refer to an entity). The attributes of the REFEX element are REG08-TYPE (name, common, pronoun, empty), and CASE (nominative, accusative, etc.).

We allow arbitrary-depth embedding of references. This means that a REFEX element may have REF element(s) embedded in it.

The second (and last) component of a GREC-ITEM is an ALT-REFEX element which is a list of REFEX elements. For the GREC tasks, these were obtained by collecting the set of all REFEXs that are in the text, and adding several defaults including pronouns and other cases (e.g. genitive) of REs already in the list.

REF elements that are embedded in REFEX elements contained in an ALT-REFEX list have an unspecified MENTION id (the '?' value). Furthermore,

such REF elements have had their enclosed REFEX removed.

The two test data sets exist in two versions:

1. Version a: each text has a single human-selected referring expression for each reference (i.e. the one found in the original Wikipedia article).
2. Version b: the same subset of texts as in (a); for this set we did not use the RES in the corpus, but replaced each of them with human-selected alternatives obtained in an online experiment as described in (Belz and Vargas, 2007); this version of the test set therefore contains three versions of each text where all the REFEXs in a given version were selected by one 'author'.

The training, development and test data for the GREC-NEG task is exactly as described above. The training and development data for the GREC-NER/Full tasks comes in two versions. The first is identical to the standard XML-annotated version of the GREC-People corpus as described above (Section 2). The second is in the test data input format.

In this format, texts have no `REFEX` and `REF` tags, and no `ALT-REFEX` element. A further difference is that in the test data format, a proportion of `REFEX` word strings have been replaced with standardised named references. All empty references have been replaced in this way, whereas (non-relative) pronouns, and previously seen named references that are not identical to the standardised named reference, are replaced with a likelihood of 0.5.

The reason for this replacement is to make both tasks easier (as we are running them for the first time) as well as more realistic (in an extractive summary, reference chains are unlikely to be as good as in the Wikipedia texts).

### 3 Evaluation Procedures

Table 3 is an overview of the evaluation measures we applied to the three tasks in GREC'10. Version a of the test sets has a single version of each text, and the scoring metrics that are based on counting matches (Word String Accuracy counts matching word strings, REG08-Type Recall/Precision count matching REG08-Type attribute values) simply count the number of matches a system achieves against that single text. Version b, however, has three versions of each text, so the match-based metrics first calculate the number of matches for each of the three versions and then use (just) the highest number of matches.

#### 3.1 Automatic Evaluations

REG08-Type Precision is defined as the proportion of `REFEXS` selected by a participating system which match the reference `REFEXS`. REG08-Type Recall is defined as the proportion of reference `REFEXS` for which a participating system has produced a match.

String Accuracy is defined as the proportion of word strings selected by a participating system that match those in the reference texts. This was computed on complete, ‘flattened’ word strings contained in the outermost `REFEX` i.e. embedded `REFEX` word strings were not considered separately.

We also computed BLEU-3, NIST, string-edit distance and length-normalised string-edit distance, all on word strings defined as for String Accuracy. BLEU and NIST are designed for multiple output versions, and for the string-edit metrics we computed the mean of means over the three text-level scores (computed against the three versions

of a text).

To measure accuracy in the NER task, we applied three commonly used performance measures for coreference resolution: MUC-6 (Vilain et al., 1995), CEAF (Luo, 2005), and B-CUBED (Bagga and Baldwin, 1998).

#### 3.2 Human-assessed evaluations

We designed the human-assessed intrinsic evaluation as a preference-judgement test where subjects expressed their preference, in terms of two criteria, for either the original Wikipedia text or the version of it with system-generated referring expressions in it. For the GREC-NEG systems, the intrinsic human evaluation involved system outputs for 30 randomly selected items from the test set. We used a Repeated Latin Squares design which ensures that each subject sees the same number of outputs from each system and for each test set item. There were three  $10 \times 10$  squares, and a total of 600 individual judgements in this evaluation (60 per system: 2 criteria  $\times$  3 articles  $\times$  10 evaluators). We recruited 10 native speakers of English from among students currently completing a linguistics-related degree at Kings College London and University College London.

For the GREC-Full systems, we used 21 randomly selected test set items, a design analogous to that for the GREC-NEG experiment, and 7 evaluators from the same cohort. This experiment had three  $7 \times 7$  squares, and 294 individual judgements.

Following detailed instructions, subjects did two practice examples, followed by the texts to be evaluated, in random order. Subjects carried out the evaluation over the internet, at a time and place of their choosing. They were allowed to interrupt and resume the experiment (though discouraged from doing so).

Figure 2 shows what subjects saw during the evaluation of an individual text pair. The place (left/right) of the original Wikipedia article was randomly determined for each individual evaluation of a text pair. People references are highlighted in yellow/orange, those that are identical in both texts are yellow, those that are different are orange (in the GREC-Full version, there were only yellow highlights). The evaluator’s task is to express their preference in terms of each quality criterion by moving the slider pointers. Moving the slider to the left means expressing a preference for

Quality criterion:	Type of evaluation:	Task:	Evaluation Method(s):
Humanlikeness	Intrinsic/automatic	NEG	1. REG'08-Type Recall and Precision 2. String Accuracy 3. String-edit distance
		NEG, Full	1. BLEU 2. NIST version of BLEU
		NER	CEAF, MUC-6, B-CUBED
Fluency	Intrinsic/human	NEG, Full	Human preference-strength judgements
Referential Clarity	Intrinsic/human	NEG, Full	Human preference-strength judgements

Table 3: Overview of GREC'10 evaluation procedures.

Exercise: GREC-NEG'09; Evaluator Jane Doe; Remaining items: 7

### Ramon Pichot Gironès

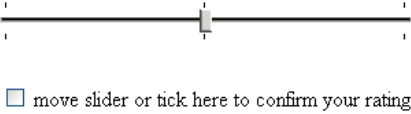
Ramon Pichot Gironès (1872 - 1 March 1925) was a Catalan and Spanish artist. He painted in an impressionist style.

He was a good friend of Pablo Picasso and acted as an early mentor to young Salvador Dalí. Salvador Dalí met Ramon Pichot Gironès in Cadaqués, Spain when Salvador was only 10 years old. Ramon also made many trips to France. Once in a while Salvador and his family would go on a trip with Ramon Pichot and his family.

Ramon Pichot Gironès (1872 - 1 March 1925) was a Catalan and Spanish artist. He painted in an impressionist style.

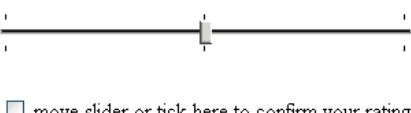
He was a good friend of Pablo Picasso and acted an early mentor to young Salvador Dalí. Salvador Dalí met him in Cadaqués, Spain when Salvador was only 10 years old. Ramon also made many trips to France. Once in a while Salvador Dalí and his family would go on a trip with Ramon Pichot and his family.

**Clarity**



move slider or tick here to confirm your rating

**Fluency**



move slider or tick here to confirm your rating

Figure 2: Example of text pair presented in human intrinsic evaluation of GREC-NEG systems.

the text on the left, moving it to the right means preferring the text on the right; the further to the left/right the slider is moved, the stronger the preference. The two criteria were explained in the introduction as follows (the wording of the first is from DUC):

1. **Referential Clarity:** It should be easy to identify who the referring expressions are referring to. If a person is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if a person is referenced, but their identity or relation to the story remains unclear.
2. **Fluency:** A referring expression should 'read well', i.e. it should be written in good, clear English, and the use of titles and names should seem natural. Note that the Fluency criterion is independent of the Referential Clarity criterion: a reference can be perfectly clear, yet not be fluent.

It was not evident to the evaluators that sliders were associated with numerical values. Slider pointers started out in the middle of the scale (no preference). The values associated with the points on the slider ranged from -10.0 to +10.0.

## 4 GREC-NEG

### 4.1 Task

The GREC-NEG test data inputs are identical to the training/development data (Figure 1), except that REF elements in the test data do not contain a REFEX element, i.e. they are 'empty'. The task for participating systems is to select one REFEX from the ALT-REFEX list for each REF in each TEXT in the test sets. If the selected REFEX contains an em-

bedded REF then participating systems also need to select a REFEX for this embedded REF and to set the value of its MENTION attribute. The same applies to all further embedded REFEXs, at any depth of embedding.

## 4.2 Systems

**NEG-Base-rand, NEG-Base-freq, NEG-Base-1st, NEG-Base-name:** We created four baseline systems each with a different way of selecting a REFEX from those REFEXs in the ALT-REFEX list that have matching entity IDs. *Base-rand* selects a REFEX at random. *Base-1st* selects the first REFEX (unless the first is the empty reference in which case it selects the second).<sup>1</sup> *Base-freq* selects the first REFEX with a REG08-TYPE and CASE combination that is the overall most frequent (as determined from the training/development data) given the SYNCAT, SYNFUNC and SEMCAT of the reference.<sup>1</sup> *Base-name* selects the shortest REFEX with attribute REG08-TYPE=name.

**UMUS:** The UMUS system maps REFEXs to class labels encoding REG08-TYPE, CASE, pronoun type, reflexiveness and recursiveness. References are represented by a set of features encoding the attributes given in the corpus, information about intervening references to other entities, preceding punctuation, sentence and paragraph boundaries, surrounding word and POS n-grams, etc. A Conditional Random Fields method is then used to map features to class labels. The problem is construed as predicting a sequence of class labels for each entity, to avoid repetition. If there is more than one REFEX available with the predicted label then the longest one is chosen the first time, and selection iterates through the list subsequently.

**UDeI:** The UDeI system is a set of decision-tree classifiers (separate ones for the main subject and other person entities) using psycholinguistically inspired features that predict the REG08-TYPE and CASE of the REFEX to select. Then the system applies rules governing the length of first and subsequent mentions. There are back-off rules for when the predicted type/case is not available. An ambiguity checker avoids the use of a pronoun if there has been an intervening reference to a person of the same gender.

**JU:** The JU baseline system is similar to our NEG-Base-freq system described above. The sub-

mitted JU system adds features to the set of REF attributes available from the corpus, including indices for paragraph, sentence and word. It also adds features to the REFEX attributes available from the corpus, in order to distinguish between several REFEXs that match the predicted REG08-TYPE and CASE combination.

**XRCE:** The XRCE system uses a conditional random field model in combination with the SampleRank algorithm for learning model parameters. The feature functions used include unary ones (>100 features encoding the attributes provided in the corpus as well as position within sentence, adjacent POS tags, etc.) and binary ones (distance to previous mention, distribution of type and case). Some binary feature functions are activated only if the previous mention was a name and control overuse of pronouns.

## 4.3 Evaluation results

Participants computed evaluation scores on the development set, using the geval code provided by us which computes Word String Accuracy, REG'08-Type Recall and Precision, string-edit distance and BLEU. The following is a summary of teams' self-reported scores:

	Recall	Precision	WSA
UMUS	0.816	0.829	0.813
UMUS'09	0.830	0.830	0.786
XRCE	0.771	0.771	0.702
UDeI	0.758	0.758	0.650
JU	0.66	0.63	0.54

REG08-Type Recall and Precision results for Test Set NEG-a (version a of the test set with just one REFEX for each REF) are shown in Table 4. As would be expected, results on the test data are somewhat worse than on the development data. Also included in this table are results for the 4 baseline systems, and it is clear that selecting the most frequent RE type and case combination given SEMCAT, SYNFUNC and SYNCAT (as done by the Base-freq system) provides a strong baseline, although it is a much better predictor for Composer and Inventor texts than Chef texts.

The last 6 columns in Table 4 contain Recall (R) and Precision (P) results for the three subdomains. For most of the systems results are slightly better for Composers than for Chefs. A contributing factor to this may be the fact that Chef texts tend to be much more colloquial. A striking detail is the collapse in scores in the Inventors subdomain for

<sup>1</sup>Note that this is a change from GREC'09.



System	REG08-Type Precision and Recall Scores against Corpus (Test Set NEG-a)																
	All						Chefs		Composers		Inventors						
	Precision			Recall			P	R	P	R	P	R					
UMUS	80.71	A					78.31	A				79.19	75.44	80.88	78.68	81.66	80.05
UMUS'09	80.17	A					77.06	A				75.16	70.71	82.25	79.54	80.66	78.08
XRCE	74.26	A					71.38	A				68.55	64.50	75.44	72.96	76.84	74.38
JU	66.98	A	B				64.38	A	B			79.56	74.85	84.32	81.55	26.97	26.11
Base-freq	61.52	A	B	C			59.60	A	B	C		51.86	49.41	65.74	63.95	62.12	60.59
UDel-NEG	60.92	A	B	C			58.56	A	B	C		55.35	52.07	62.43	60.37	62.85	60.84
Base-rand	43.32		B	C			42.00		B	C		40.43	38.76	43.00	41.77	46.21	45.07
Base-name	40.60			C			39.09			C		47.80	44.97	40.32	39.06	35.28	34.24
Base-1st	40.25			C			39.64			C		47.88	46.75	39.71	39.20	34.91	34.48

Table 4: REG08-Type Precision and Recall scores against corpus version of Test Set for complete set and for subdomains; homogeneous subsets (Tukey HSD, alpha = .05) for complete set only.

System	REG08-Type Precision and Recall Scores against human topline (Test Set NEG-b)																
	All						Chefs		Composers		Inventors						
	Precision			Recall			P	R	P	R	P	R					
Corpus	82.67	A					84.01	A				82.25	84.24	83.26	84.47	82.02	83.04
UMUS	81.64	A					80.49	A				82.92	80.91	80.59	79.54	82.41	81.80
UMUS'09	80.46	A					78.59	A	B			80.50	77.58	80.62	79.10	80.15	78.55
XRCE	73.76	A	B				72.04	A	B	C		73.58	70.91	74.11	72.71	73.28	71.82
UDel-NEG	65.54	A	B	C			64.01	A	B	C	D	66.04	63.64	66.12	64.88	64.12	62.84
Base-freq	65.38	A	B	C			64.37	A	B	C	D	59.94	58.48	68.97	68.07	63.64	62.84
JU	63.73	A	B	C			62.25	A	B	C	D	76.42	73.64	76.04	74.60	32.32	31.67
Base-name	55.22		B	C			54.01		B	C	D	56.29	54.24	58.05	57.04	49.49	48.63
Base-1st	54.68		B	C			54.68		B	C	D	55.45	55.45	57.68	57.68	48.88	48.88
Base-rand	48.46			C			47.75			C	D	48.77	47.88	47.13	46.44	50.51	49.88

Table 5: REG08-Type Recall and Precision scores against human topline version of Test Set for complete set and for subdomains; homogeneous subsets (Tukey HSD, alpha = .05) for complete set only.

the JU system. As a side effect, the resulting variation led to fewer significant differences between systems being found in the results than would have been the case otherwise.

We carried out univariate ANOVAs with System as the fixed factor, and REG08-Type Recall as the dependent variable in one ANOVA, and REG08-Type Precision in the other. The F-ratio for Recall was  $F_{(9,990)} = 13.253, p < 0.001$ .<sup>2</sup> The F-ratio for Precision was  $F_{(9,990)} = 12.670, p < 0.001$ . The columns containing single capital letters in Table 4 show the homogeneous subsets of systems as determined by a post-hoc Tukey HSD analysis. Systems whose scores are not significantly different (at the .05 level) share a letter.

Table 5 shows analogous results computed against Test Set NEG-b (which has three versions of each text). Table 5 includes results for the corpus texts, also computed against the three versions of each text in test set GREC-NEG-b. We performed univariate ANOVAs with System as the fixed factor, and Recall as the dependent variable in one, and Precision in the other. The result for Recall was  $F_{(9,990)} = 5.248, p < .001$ , and for Precision  $F_{(9,990)} = 5.038, p < .001$ . We again compared the mean scores with Tukey's HSD.

<sup>2</sup>We included the corpus texts themselves in the analysis, hence 9 degrees of freedom (10 systems).

One would generally expect results on test set NEG-b to be better than on NEG-a. This is the case for all baseline systems and some of the participating systems, but not all. The JU system in particular drops in score (and rank).

We also computed Word String Accuracy and the other string similarity metrics described in Section 3 for the GREC-NEG Task. The resulting scores for Test Set NEG-a are shown in Table 6. Ranks for peer systems relative to each other are very similar to the results for REG08-Type reported above.

We performed a univariate ANOVA with System as the fixed factor, and Word String Accuracy as the dependent variable. The F-ratio for System was  $F_{(9,990)} = 41.308, p < 0.001$ ; the homogeneous subsets resulting from the Tukey HSD post-hoc analysis are shown in columns 3–7 of Table 6.

Table 7 shows analogous results for human topline Test Set NEG-b (which has three versions of each text). We carried out the same kind of ANOVA as for Test Set NEG-a; the result for System on Word String Accuracy was  $F_{(9,990)} = 35.123, p < 0.001$ . System rankings are the same as for Test Set NEG-a (the differences between JU and Base-freq, which swap ranks, are not significant); scores across the board (again, except for the JU system) are somewhat higher, because of the way scores are computed for version b test

System	String similarity against Corpus (Test Set NEG-a)												
	Word String Accuracy									BLEU-3	NIST	SE	norm. SE
	All					Chefs	Composers	Inventors					
UMUS	78.51	A					76.42	79.29	78.88	0.7968	7.4986	0.6063	0.2019
UMUS'09	75.05	A					69.18	77.66	75.32	0.7615	6.9865	0.6806	0.2233
XRCE	65.25	A					61.01	66.12	67.18	0.7031	6.0264	0.8969	0.3131
JU	60.71	A					72.96	76.63	23.41	0.5720	5.7264	1.1810	0.3671
Base-freq	57.10	A	B				50.31	60.65	56.49	0.5913	4.9860	1.2249	0.4191
UDeL-NEG	38.21		B	C			37.42	39.20	37.15	0.5498	5.0211	1.6222	0.5869
Base-name	28.48			C	D		35.53	27.51	24.43	0.4966	4.9355	1.8017	0.6662
Base-rand	8.22				D	E	8.49	7.10	9.92	0.1728	1.2501	2.4290	0.8928
Base-1st	4.69					E	3.46	5.47	4.33	0.1990	2.4018	2.9906	0.8152

Table 6: Word String Accuracy, BLEU, NIST, and string-edit scores, computed on Test Set NEG-a (systems in order of Word String Accuracy); homogeneous subsets (Tukey HSD, alpha = .05) for String Accuracy only.

System	String similarity against human topline (Test Set NEG-b)												
	Word String Accuracy									BLEU-3	NIST	SE	norm. SE
	All					Chefs	Composers	Inventors					
Corpus	81.90	A					83.33	82.25	80.15	0.9499	9.1087	0.7082	0.2517
UMUS	77.29	A	B				79.25	76.48	77.10	0.9296	8.1746	0.8383	0.2906
UMUS'09	74.84	A	B	C			73.58	75.59	74.55	0.8968	7.5005	0.9096	0.3083
XRCE	63.95	A	B	C			66.35	63.02	63.61	0.7960	6.0780	1.1577	0.4060
Base-freq	59.84		B	C	D		55.97	62.72	58.02	0.7393	5.4920	1.3949	0.4717
JU	56.31			C	D	E	68.87	66.86	27.99	0.5765	5.8764	1.5114	0.4720
UDeL-NEG	41.60				D	E	44.34	40.38	41.48	0.6503	5.9571	1.7138	0.6057
Base-name	37.27					E	42.14	36.83	34.10	0.6480	6.6551	1.7299	0.6287
Base-rand	10.45						10.06	9.91	11.70	0.2468	1.4828	2.4869	0.8884
Base-1st	8.58					F	5.66	10.95	6.87	0.2824	3.5790	2.9226	0.7868

Table 7: Word String Accuracy, BLEU, NIST, and string-edit scores, computed on Test Set NEG-b (systems in order of Word String Accuracy); homogeneous subsets (Tukey HSD, alpha = .05) for String Accuracy.

sets: a score is the highest score a system achieves (at text-level) against any of the three versions of a test set text that is taken into account.

Results for BLEU-3, NIST and the two string-edit distance metrics are shown in the rightmost 4 columns of Tables 6 and 7. With the exception of Base-freq/Basename on Test Set NEG-b, systems whose Word String Accuracy scores differ significantly are assigned the same relative ranks by all other string-similarity metrics as by Word String Accuracy.

In the human intrinsic evaluation, evaluators rated system outputs in terms of whether they preferred them over the original Wikipedia texts. As a result of the experiment we had (for each system and each evaluation criterion) a set of scores ranging from -10.0 to +10.0, where 0 meant no preference, negative scores meant a preference for the Wikipedia text, and positive scores a preference for the system-produced text.

The second column of the left half of Table 8 summarises the Clarity scores for each system in terms of their mean; if the mean is negative the evaluators overall preferred the Wikipedia texts, if it is positive evaluators overall preferred the system. The more negative the score, the more strongly evaluators preferred the Wikipedia texts.

Columns 8–10 show corresponding counts of how many times each system was preferred (+), dispreferred (−), and neither (0).

The other half of Table 8 shows corresponding results for Fluency.

We ran a factorial multivariate ANOVA with Fluency and Clarity as the dependent variables. In the first version of the ANOVA, the fixed factors were System, Evaluator and Wikipedia\_Side (indicating whether the Wikipedia text was shown on the left or right during evaluation). This showed no significant effect of Wikipedia\_Side on either Fluency or Clarity, and no significant interaction between any of the factors. There was also no significant effect of Evaluator on Fluency, and only a weakly significant effect of Evaluator on Clarity. We ran the ANOVA again, this time with just System as the fixed factor. The F-ratio for System on Fluency was  $F_{(9,290)} = 22.911, p < .001$ , and for System on Clarity it was  $F_{(9,290)} = 13.051, p < .001$ . Post-hoc Tukey's HSD tests revealed the significant pairwise differences indicated by the letter columns in Table 8.

Correlation between individual Clarity and Fluency ratings as estimated with Pearson's coefficient was  $r = 0.66, p < 0.01$ , indicating that the two criteria covary to some extent.

Clarity						Fluency												
System	Mean					+	0	-	System	Mean					+	0	-	
Corpus	0.000	A				1	28	1	Corpus	0.133	A				1	29	0	
UMUS	-2.023	A	B			1	13	16	UMUS	-1.640	A	B			4	12	14	
UMUS'09	-2.527	A	B	C		0	15	15	UMUS'09	-2.130	A	B			3	11	16	
Base-name	-2.900		B	C		1	7	22	XRCE	-3.587		B	C		2	8	20	
Base-1st	-3.160		B	C		4	3	23	JU	-4.057		B	C	D	0	10	20	
XRCE	-3.500		B	C	D	1	9	20	Base-freq	-4.990			C	D	1	3	26	
JU	-3.577		B	C	D	0	10	20	Base-name	-6.620				D	E	0	1	29
UDel-NEG	-5.137			C	D	E	0	1	29	Base-1st	-7.823				E	1	0	29
Base-freq	-6.190				D	E	0	2	28	Base-rand	-7.950				E	1	0	29
Base-rand	-7.663					E	1	0	29	UDel-NEG	-7.970				E	0	1	29

Table 8: GREC-NEG: Results for Clarity and Fluency preference judgement experiment. Mean = mean of individual scores (where scores ranged from -10.0 to + 10.0); + = number of times system was preferred; - = number of times corpus text (Wikipedia) was preferred; 0 = number of times neither was preferred.

The relative ranks of the peer systems are the same in terms of both Fluency and Clarity. However, there are interesting differences in the ranks of the baseline systems. For Clarity, Base-name and Base-1st are scored fairly highly (presumably because both tend to pick named references which are clear if not always fluent), but both go back to not being significantly better than Base-rand in the Fluency rankings. Base-freq does badly in the Clarity scores, but is significantly better than the bottom three systems in terms of Fluency.

## 5 GREC-NER

### 5.1 Task

The GREC-NER task is a straightforward combined named-entity recognition and coreference resolution task, restricted to people entities. The aim for participating systems is to identify all those types of mentions of people that we have annotated in the GREC-People corpus, and to insert REF and REFEX tags with coreference IDs into the texts.

### 5.2 Systems

**Baselines:** We used the coreference resolvers included in the LingPipe<sup>3</sup> and OpenNLP Tools<sup>4</sup> packages as baseline systems.

**Poly-co:** The Poly-co system starts by applying a POS tagger to the input text. A Conditional Random Fields classifier (trained on an automatically annotated Wikipedia corpus) is then used to detect named mentions, using word and POS based features. Logical rules then detect pronoun mentions, using named-entity, word and POS features. Coreference of named mentions is determined by clustering with a similarity measure based on words, POS tags and sentence position,

<sup>3</sup><http://alias-i.com/lingpipe/>

<sup>4</sup><http://opennlp.sf.net>

applied to mentions in order of their appearance. Coreference of pronouns is determined with the Hobbs algorithm for anaphora resolution.

**UDel-NER:** The UDel-NER system starts by (1) parsing the input text with the Stanford Parser, from which it extracts syntactic functions of words and relationships between them; and (2) separately applying the Stanford Named Entity Recognizer. Pronoun and common noun mentions are identified using lists of all English pronouns and of common nouns which could conceivably be used to refer to people (occupations like ‘painter’, family relations like ‘grandmother’, etc.). Values for all REF and REFEX attributes except coreference ID are obtained. Finally, the system applies a coreference resolution tool which compares each reference to all previous references in reverse order, on the basis of case, gender, number, syntactic function, and REG’08-Type.

### 5.3 Results

The coreference resolution accuracy scores for the GREC-NER systems are shown in Table 9. The two participating systems are both significantly better than the two baselines in terms of their mean coreference resolution accuracy scores.

## 6 GREC-Full

### 6.1 Task

The aim for GREC-Full systems was to improve the referential clarity and fluency of input texts. Participants were free to do this in whichever way they chose. Participants were encouraged, though not required, to create systems which replace referring expressions as and where necessary to produce as clear and fluent a text as possible. This task could be viewed as composed of three sub-tasks: (1) named entity recognition (as in GREC-

	Test set				
	Mean		B-3	CEAF	MUC
UDeI-NER	72.71	A	80.51	77.53	60.09
Poly-co	66.99	A	76.92	70.29	53.77
LingPipe	58.23		71.19	61.58	41.92
OpenNLP	54.03		67.61	59.17	35.32

Table 9: MUC-6, CEAF and B-3 scores for GREC-NER systems. Systems shown in order of average scores.

NER); (2) a conversion tool to give lists of possible referring expressions for each entity; and (3) named entity generation (as in GREC-NEG).

## 6.2 Systems

All GREC-Full systems in our evaluations are composed of a GREC-NER and a GREC-NEG system. We created three baseline systems. Two of these we created by combining the two GREC-NER baseline systems with the random GREC-NEG baseline system (Base-rand). For this purpose we created a simple conversion utility which adds default REFEXS. The third baseline system combines the UDeI-NER system with Base-rand.

The only team that submitted both a GREC-NER and a GREC-NEG system was UDeI. All other GREC-Full systems therefore combine the efforts of two teams (for overview of system combinations, please refer to Table 1). The two system combinations involving the UDeI-NER system did not require a conversion utility, because UDeI-NER already outputs full GREC-People format.

## 6.3 Results

NIST and BLEU scores computed against the Wikipedia texts for the GREC-Full systems are shown in Table 10. Note that these have been computed on the complete texts, not just the referential expressions (which explains the high BLEU scores). The scores in the second row (Corpus, test set vers.) are obtained by comparing the test set versions of the corpus texts (in which some of the references have been replaced with standardised named references, as explained in Section 2) against the Wikipedia texts. The two halves of the table show scores computed against version a of the test set (the original Wikipedia texts) on the left, and against version b of the test set (which has three versions of each text with human-selected RES) on the right.

In the human intrinsic evaluation of GREC-Full systems, evaluators again rated system outputs in terms of whether they preferred them over the

original Wikipedia texts. Table 11 shows the results in the same format as in Table 8 for the GREC-NEG systems.

We ran the same two factorial multivariate ANOVAs with Fluency and Clarity as the dependent variables. In the first version of the ANOVA, there were no effects of Evaluator (apart from a mild one on Clarity) and Wikipedia\_Side and no significant interaction between any of the factors. There was no effect of Evaluator on Fluency and only a mild effect of Evaluator on Clarity. The second ANOVA just had System as the fixed factor. The F-ratio for Fluency was  $F_{(6,140)} = 13.054, p < .001$ , and for System on Clarity it was  $F_{(6,140)} = 14.07, p < .001$ . Post-hoc Tukey’s HSD tests revealed the significant pairwise differences indicated by the letter columns in Table 11.

Correlation between individual Clarity and Fluency ratings as estimated with Pearson’s coefficient was  $r = 0.696, p < .01$ , indicating that the two criteria covary to some extent.

Apart from UDeI-Full and OpenNLP/Base-rand switching places, system ranks are the same for Fluency and Clarity. Moreover, system ranks are very similar to those produced by the string-similarity scores above. UDeI-Full is a much harder task than GREC-NEG and it is a very good result indeed for a system to be preferred over Wikipedia once or twice and to be rated equally good as Wikipedia 4–7 times.

## 7 Concluding Remarks

GREC’10 has, for the first time, produced systems which can do end-to-end named-entity generation, moreover most of which can do it well enough for human judges do rate them as good as Wikipedia or better around one third of the time.

This was the second time the GREC-NEG Task was run, and the first time GREC-NER and GREC-Full were run. As in 2009, many more teams registered than were able to submit a system by the deadline, but we hope that the GREC data (which is now freely available) will lead to many more re-

Test Set NEG-Full-a							Test Set NEG-Full-b								
System	Mean text-level BLEU-4					BLEU-4	NIST	System	Mean text-level BLEU-4					BLEU-4	NIST
Corpus	1.00	A				1.000	13.71	Corpus	.991	A				0.985	13.74
Corpus (test set vers.)	.941		B			0.923	12.92	Corpus (test set vers.)	.946		B			0.929	13.20
UDeI/UMUS	.934		B	C		0.925	13.13	UDeI/UMUS	.939		B	C		0.928	13.29
UDeI/XRCE	.921		B	C		0.898	12.98	UDeI/XRCE	.928		B	C		0.907	13.15
UDeI-Full	.905			C		0.870	12.59	UDeI-Full	.912			C		0.882	12.82
UDeI/Base-rand	.812				D	0.809	12.17	UDeI/Base-rand	.823				D	0.821	12.43
OpenNLP/Base-rand	.809				D	0.775	11.49	OpenNLP/Base-rand	.817				D	0.785	11.72
LingPipe/Base-rand	.752				E	0.753	11.48	LingPipe/Base-rand	.763				E	0.764	11.70

Table 10: GREC-FULL: Mean text-level BLEU-4 scores, system-level BLEU-4 and NIST scores.

Clarity							Fluency										
System	Mean					+	0	-	System	Mean					+	0	-
Corpus	-0.033	A				1	20	0	Corpus	0	A				0	30	0
UDeI/XRCE	-2.209	A	B			0	6	15	UDeI/XRCE	-3.424		B			1	4	16
UDeI/UMUS	-2.638	A	B			1	6	14	UDeI/UMUS	-4.057		B	C		2	5	14
UDeI-Full	-2.833		B			0	7	14	OpenNLP/Base-rand	-4.671		B	C		2	4	15
OpenNLP/Base-rand	-3.486		B			1	7	13	UDeI-Full	-4.967		B	C		0	4	16
UDeI/Base-rand	-4.667		B			0	5	16	UDeI/Base-rand	-6.800			C	D	0	2	19
LingPipe/Base-rand	-7.829			C		0	0	21	LingPipe/Base-rand	-8.405				D	0	0	21

Table 11: GREC-FULL: Results for Clarity and Fluency preference judgement experiment. Mean = mean of individual scores (where scores ranged from -10.0 to + 10.0); + = number of times system was preferred; - = number of times corpus text (Wikipedia) was preferred; 0 = number of times neither was preferred.

sults being produced and reported over time.

## Acknowledgments

Many thanks to the members of the Corpora and SIGGEN mailing lists, and Brighton University colleagues who helped with the online RE selection experiments for the b-versions of the test sets. Thanks are also due to the Oxford, Kings College London and University College London students who helped with the intrinsic evaluation experiments.

## References

- A. Bagga and B. Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at LREC'98*, pages 563–566.
- A. Belz and S. Varges. 2007. Generation of repeated references to discourse entities. In *Proceedings of ENLG'07*, pages 9–16.
- A. Belz. 2010. GREC named entity recognition and GREC named entity regeneration challenges 2010: Participants' Pack. Technical Report NLTG-10-01, Natural Language Technology Group, University of Brighton.
- R. Huddleston and G. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press.
- X. Luo. 2005. On coreference resolution performance metrics. *Proc. of HLT-EMNLP*, pages 25–32.

- M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of MUC-6*, pages 45–52.

# Named Entity Generation using Sampling-based Structured Prediction

**Guillaume Bouchard**

Xerox Research Centre Europe

6 Chemin de Maupertuis

38240 Meylan, France

guillaume.bouchard@xerox.com

## Abstract

The problem of Named Entity Generation is expressed as a conditional probability model over a structured domain. By defining a factor-graph model over the mentions of a text, we obtain a compact parameterization of what is learned using the SampleRank algorithm.

## 1 Introduction

This document describes the participation of the Xerox Research Centre Europe team in the GREC-NEG'10 challenge (<http://www.nltg.brighton.ac.uk/research/genchal10/grec/>)

## 2 Model

Conditional random fields are conditional probability models that define a distribution over a complex output space. In the context of the Named-Entity Generation challenge, the output space is the set of possible referring expressions for all the possible mentions of the text. For example, assuming that we have the following text with holes (numbers are entity IDs):

#1 was a Scottish mathematician, son of #2. #1 is most remembered as the inventor of logarithms and Napier's bones.

Then the possibilities associated with the entity #1 are:

1. John Napier of Merchistoun,
2. Napier,
3. he,
4. who,

and the possibilities associated with the entity #2 are:

1. Sir Archibald Napier of Merchiston,
2. he,
3. who.

Then, the output space is  $Y = \{1, 2, 3, 4\} \times \{1, 2, 3\} \times \{1, 2, 3, 4\}$ , representing all the possible combination of choices for the mentions. The solution  $y = (1, 1, 3)$  corresponds to inserting the texts 'John Napier of Merchiston', 'Sir Archibald Napier of Merchiston' and 'he' in the holes of the text in the same order. This is the combination that is the closest to the original text, but a human could also consider that solution  $y = (1, 1, 2)$  as being equally valid.

Denoting  $x$  the input, i.e. the text with the typed holes, the objective of the task is to find the combination  $y \in Y$  that is as close as possible to natural texts.

We model the distribution of  $y$  given  $x$  by a factor graph:  $p(y|x) \propto \prod_{c \in C} \phi_c(x, y)$ , where  $C$  is the set of factors defined over the input and output variables. In this work, we considered 3 types of exponential potentials:

- Unary potentials defined on each individual output  $y_i$ . They include more than 100 features corresponding to the position of the mention in the sentence, the previous and next part of speech (POS), the syntactic category and function of the mention, the type and case of the corresponding referring expression, etc.
- Binary potentials over contiguous mentions include the distance between them, and the joint distribution of the types and cases.
- Binary potentials that are activated only between mentions and the previous time the

same entity was referred to by a name. The purpose of this is to reduce the use of pronouns referring to a person when the mentions are distant to each other.

To learn the parameter of the factor graph, we used the SampleRank algorithm (Wick et al., 2009) which casts the prediction problem as a stochastic search algorithms. During learning, an optimal ranking function is estimated.

### 3 Results

Using the evaluation software supplied by the GREC-NEG organizers, we obtained the following performances:

total slots	: 907
reg08 type matches	: 693
reg08 type accuracy	: 0.764057331863286
reg08 type matches including embedded	: 723
reg08 type precision	: 0.770788912579957
reg08 type recall	: 0.770788912579957
total peer REFs	: 938
total reference REFs	: 938
string matches	: 637
string accuracy	: 0.702315325248071
mean edit distance	: 0.724366041896362
mean normalised edit distance	: 0.279965348873838
BLEU 1 score	: 0.7206
BLEU 2 score	: 0.7685
BLEU 3 score	: 0.7702
BLEU 4 score	: 0.754
NIST score	: 5.1208

### References

Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. 2009. SampleRank: Learning preferences from atomic gradients. *Neural Information Processing Systems (NIPS) Workshop on Advances in Ranking*.





# Poly-co : an unsupervised co-reference detection system

Éric Charton, Michel Gagnon, Benoit Ozell

École Polytechnique de Montréal

2900 boulevard Edouard-Montpetit, Montreal, QC H3T 1J4, Canada.

{eric.charton, michel.gagnon, benoit.ozell}@polymtl.ca

## Abstract

We describe our contribution to the Generation Challenge 2010 for the tasks of Named Entity Recognition and co-reference detection (GREC-NER). To extract the NE and the referring expressions, we employ a combination of a Part of Speech Tagger and the Conditional Random Fields (CRF) learning technique. We finally experiment an original algorithm to detect co-references. We conclude with discussion about our system performances.

## 1 Introduction

Three submission tracks are proposed in Generation Challenges 2010. **GREC-NEG**, where participating systems select a referring expression (RE) from a given list. **GREC-NER** where participating systems must recognize all mentions of people in a text and identify which mentions co-refer. And **GREC-Full**, end-to-end RE regeneration task; participating systems must identify all mentions of people and then aim to generate improved REs for the mentions. In this paper we present an unsupervised CRF based Named Entity Recognition (NER) system applied to the **GREC-NER** Task.

## 2 System description

The proposed system follows a pipelined architecture (each module processes the information provided by the previous one). First, a *Part of Speech* (POS) tagger is applied to the corpus. Then, the combination of words and POS tags are used by a CRF classifier to detect *Named Entities* (NE). Next, logical rules based on combination of POS tags, words and NE labels are used to detect pronouns related to *persons*. Finally, an algorithm

identifies, among the *person* entities that have been detected, the ones that co-refer and cluster them. At the end, all collected information is aggregated in a XML file conform to **GREC-NER** specifications.

### 2.1 Part of speech

The part of speech labeling is done with the English version of Treetagger<sup>1</sup>. It is completed by a step where every *NAM* tag associated to a first name is replaced by a *FNAME* tag, using a lexical resource of first names (see table 2, column *POS Tag*). The first name tag improves the NE detection model while it improves the estimation of conditional probabilities for words describing a person, encountered by a NER system.

Word from Corpus	POS Tag	NE Tag
Adrienne	FNAM	PERS
Calvo	NAM	PERS
enrolled	VVD	UNK
at	IN	UNK
Johnson	NAM	ORG
Wales	NAM	ORG
College	NAM	ORG

Table 2: Sample of word list with POS Tagging and NE tagging

### 2.2 Named entity and pronoun labeling

The Named Entity Recognition (NER) system is an implementation of the CRF based system (Béchet and Charton, 2010) that has been used in the French NER evaluation campaign ESTER 2 (Galliano et al., 2009)<sup>2</sup>. For the present task, training of the NER tool is fully unsupervised as it does not use the GREC training corpus. It is trained in English with an automatically NE annotated version of the Wikipedia Corpus (the full system configuration is described in (Charton and

<sup>1</sup>The Tree-tagger is a tool for annotating text with part-of-speech and lemma information. <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

<sup>2</sup>Referenced in this paper as *LIA*

<sup>1</sup>This work is granted by Unima Inc and Prompt Québec

Poly-co Score	B3			CEAF			MUC		
	Precision	Recall	FScore	Precision	Recall	FScore	Precision	Recall	FScore
Full set	91.48	85.89	<b>88.60</b>	85.40	85.40	<b>85.40</b>	92.15	86.95	<b>89.47</b>
Chef	91.12	87.84	<b>89.45</b>	86.53	86.53	<b>86.53</b>	91.86	88.55	<b>90.18</b>
Composers	92.01	87.14	<b>89.51</b>	86.87	86.87	<b>86.87</b>	92.11	87.02	<b>89.49</b>
Inventors	91.27	82.63	<b>86.74</b>	82.73	82.73	<b>82.73</b>	92.48	85.29	<b>88.74</b>

Table 1: System results obtained on dev-set

Torres-Moreno, 2010)). It is able to label PERS<sup>3</sup>, ORG, LOC, TIME, DATE. We measured a specific precision of 0,93 on PERS NE detection applied to the English ACE<sup>4</sup> evaluation set.

Following the NE detection process, detection rules are used to label each personal pronoun with the PERS tag. Boolean AND rules are applied to triples  $\{word, POS\ tag, NE\ tag\}$ , where  $word = \{he, him, she, her \dots\}$ ,  $POS\ tag=NN$ , and  $NE\ tag=UNK$ . This rule structure is adopted to avoid the PERS labeling of pronouns included in an expression or in a previously tagged NE (i.e a music album or a movie title, using word *She*, and previously labeled with PROD NE tag). Finally, each PERS labeled entity is numbered by order of apparition and is associated with the sentences reference number where it appears (consecutive PERS labeled words, not separated by punctuation mark, receive the same index number).

### 2.3 Entities clustering by unstacking

In the final stage, our system determines which entities co-refer. First, a clustering process is achieved. The principle of the algorithm is as follows: entities characteristics (words, POS tags, sentence position) are indexed in a stack, ordered according to their chronological apparition in the text (the entity at the top of the stack is the first one that has been detected in the document). At the beginning of the process, the entity that is at the top of the stack is removed and constitutes the first item of a cluster. This entity is compared sequentially, by using similarity rules, with every other entities contained in the stack. When there is a match, entity is transferred to the currently instantiated cluster and removed from the stack. When the end of the stack is reached, remaining entities are reordered and the process iterates from the beginning. This operation is repeated until the stack is empty.

Comparison of entities in the stack is done in

<sup>3</sup>PERS tag is commonly used in NER Task to describe labels applied to people, ORG describe organisations, LOC is for places.

<sup>4</sup>ACE is the former NIST NER evaluation campaign.

two ways according to the nature of the entity. We consider a candidate entity  $E_c$  from stack  $S$ . According to iteration  $k$ , the current cluster is  $C_k$ . Each element of the sequence  $E_c$  (i.e *Chester FNAME Carton NAM*) is compared to the sequences previously transferred in  $C_k$  during the exploration process of the stack. If  $E_c \subseteq \bigcup C_k$ , it is included in cluster  $C_k$  and removed from  $S$ . Finally inclusion of pronouns from  $S$  in  $E_c$  is done by resolving the anaphora, according to the Hobbs algorithm, as described in (Jurafsky et al., 2000)<sup>5</sup>.

### 3 Results and conclusions

Table 1 shows our results on dev-set. We obtain good precision on the 3 subsets. Our system slightly underperforms the recall. This can be explained by a good performance in the NE detection process, but a difficulty in some cases for the clustering algorithm to group entities. We have observed in the Inventors dev-set some difficulties, due to strong variation of surface forms for specific entities. We plan to experiment the use of an external resource of surface forms for person names extracted from Wikipedia to improve our system in such specific case.

### References

- Frédéric Béchet and Eric Charton. 2010. Unsupervised knowledge acquisition for extracting named entities from speech. In *ICASSP 2010*, Dallas. ICASSP.
- Eric Charton and J.M. Torres-Moreno. 2010. NL-GbAse: a free linguistic resource for Natural Language Processing systems. In *LREC 2010*, editor, *English*, number 1, Matla. Proceedings of LREC 2010.
- S. Galliano, G. Gravier, and L. Chaubard. 2009. The ESTER 2 Evaluation Campaign for the Rich Transcription of French Radio Broadcasts. In *International Speech Communication Association conference 2009*, pages 2583–2586. Interspeech 2010.
- D. Jurafsky, J.H. Martin, A. Kehler, K. Vander Linden, and N. Ward. 2000. *Speech and language processing*. Prentice Hall New York.

<sup>5</sup>p704, 21.6

# JU\_CSE\_GREC10: Named Entity Generation at GREC 2010

Amitava Das<sup>1</sup>, Tanik Saikh<sup>2</sup>, Tapabrata Mondal<sup>3</sup>, Sivaji Bandyopadhyay<sup>4</sup>

Department of Computer Science and Engineering  
Jadavpur University,  
Kolkata-700032, India

amitava.santu@gmail.com<sup>1</sup>, tanik4u@gmail.com<sup>2</sup>, tapabratamondal@gmail.com<sup>3</sup>, sivaji\_cse\_ju@yahoo.com<sup>4</sup>

## Abstract

This paper presents the experiments carried out at Jadavpur University as part of the participation in the GREC Named Entity Generation Challenge 2010. The Baseline system is based on the SEMCAT, SYNCAT and SYNFUNC features of REF and REG08-TYPE and CASE features of REFEX elements. The discourse level system is based on the additional positional features: paragraph number, sentence number, word position in the sentence and mention number of a particular named entity in the document. The inclusion of discourse level features has improved the performance of the system.

## 1 Baseline System

The baseline system is based on the following linguistic features of REF elements: SEMCAT (Semantic Category), SYNCAT (Syntactic Category) and SYNFUNC (Syntactic Function) (Anja Belz, 2010) and the following linguistic features of REFEX elements: REG08-TYPE (Entity type) and CASE (Case marker). The baseline system has been separately trained on the training set data for the three domains: chefs, composers and inventors. The system has been tested on each development set by identifying the most probable REFEX element among the possible alternatives based on the REF element feature combination. The probability assigned to a REFEX element corresponding to a certain feature combination of REF element is calculated as follows:

$$p(R_v) = \frac{N_{REFEX}^{D_i}}{N_{REF}^{D_i}}$$

where  $p(R_v)$  is the probability of the targeted REFEX element to be assigned,  $N_{REF}^{D_i}$  is the total number of occurrences of REF element feature combinations,  $D_i$  denotes the domain i.e., Chefs,

Composers and Inventors and  $N_{REF}^{D_i}$  denotes the total number of occurrences of the REFEX element corresponding to the REF feature combination.

It has been observed that many times the most probable REFEX element as identified from the training set is not present among the alternative REFEX elements. In these cases the system assigns the next highest probable REFEX element learnt from the training set that matches with one of the REFEX elements among the alternatives. In some cases more than one REFEX element get same probability in the training set. In these cases, the REFEX element that occurs earlier in the alternative set is assigned. The experimental result of Baseline system is reported in Table 1.

	Chefs	Composers	Inventors
<b>Precision</b>	0.63	0.68	0.70
<b>Recall</b>	0.69	0.60	0.64
<b>F-Measure</b>	0.66	0.64	0.68

Table 1: Result of Baseline System

## 2 Discourse Level System

The discourse level features like paragraph number, sentence number and position of a particular word in a sentence have been added with the features considered in the baseline system. As mentioned in Section 1, more than one REFEX element can have the same probability value. This happens as REFEX elements are identified by two features only REG08-TYPE and CASE.

	Nam e	Pro- noun	Com- mon	Emp- ty
Chefs	2317	3071	55	646
Composers	2616	4037	92	858
Inventors	1959	2826	75	621

Table 2: Distribution of REFEX Types among three domains.

The above problem occurs mainly for *Name* type. Pronouns are very frequent in all the three domains but they have small number of variations as: he, her, him, himself, his, she, who, whom and whose. Common type REFEX ele-

ments are too infrequent in the training set and they are very hard to generalize. *Empty* type has only one REFEX value as: “\_”. The distribution of the various REFEX types among the three domains in the training set is shown in Table 2.

## 2.1 Analysis of Name type entities

Table 2 shows that *name* types are very frequent in all the three domains. *Name* type entities are further differentiated by adding more features derived from the analysis of the *name* type element.

Firstly, the full name of each named entity has been identified by Entity identification number (id), maximum length among all occurrences of that named entity and case marker as plain. For example, in Figure 1, the REFEX element of id 3 has been chosen as a full name of entity “0” as it has the longest string with case “plain”.

After identification of full name of each REFEX entity, the following features are identified for each occurrence of an entity:: Complete Name Genitive (CNG), Complete Name (CN), First Name Genitive (FNG), First Name (FN), Last Name Genitive (LNG), Last Name (LN), Middle Name Genitive (MNG) and Middle Name (MN). These features are binary in nature and for each occurrence of an entity only one of the above features will be true.

Pronouns are kept as the REFEX element feature with its surface level pattern as they have

only 9 variations. Common types are considered with tag level “common” as they hard to generalize. Empty types are tagged as “empty” as they have only one tag value “\_”.

1	<REFEX ENTITY="0" REG08-TYPE="name" CASE="genitive">Alain Senderens's</REFEX>	CNG
2	<REFEX ENTITY="0" REG08-TYPE="name" CASE="genitive">Senderens's</REFEX>	LNG
3	<REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Alain Senderens</REFEX>	CN
4	<REFEX ENTITY="0" REG08-TYPE="name" CASE="plain">Senderens</REFEX>	LN

Figure 1: Example of Full Name Identification

## 3 Experimental Results

The experimental results of the discourse level system on the development set are reported in the Table 3 and Table 4 respectively. Table 3 reports the results when the system has been trained separately with domain specific training set and Table 4 reports the results when the training has been carried out on the complete training set.

The comparison of the results of the baseline and the discourse level system shows an overall improvement. But there are some interesting observations when comparing the results in Table 3 and Table 4. Currently detailed analyses of the results are being carried out.

	Chefs			Composers			Inventors		
	P	R	F	P	R	F	P	R	F
<b>Name</b>	0.69	0.74	0.71	0.78	0.61	0.69	0.77	0.67	0.71
<b>Pronoun</b>	0.81	0.76	0.79	0.70	0.84	0.76	0.76	0.87	0.81
<b>Common</b>	0.76	0.87	0.81	0.37	0.44	0.40	0.44	0.65	0.68
<b>Empty</b>	0.92	0.88	0.90	0.86	0.92	0.89	0.72	0.65	0.68

Table 3: Experimental Results of Discourse Level System on the Development Set (Training with Domain Specific Training Set)

	Reg08 Type		String Accuracy	BLEU				NIST	String Edit Distance	
	Precision	Recall		1	2	3	4		Mean	Mean Normalized
<b>Composers</b>	0.63	0.67	0.56	0.61	0.57	0.54	0.50	3.34	1.07	0.40
<b>Inventors</b>	0.60	0.62	0.50	0.55	0.54	0.52	0.49	2.90	1.25	0.47
<b>Total</b>	0.63	0.66	0.54	0.61	0.58	0.57	0.55	3.83	1.03	0.42

Table 4: Table 4: Experimental Results of Discourse Level System on the Development Set (Training with Complete Training Set)

## References

Anja Belz. 2010. GREC Named Entity Generation Challenge 2010: Participants' Pack.

# The UMUS system for named entity generation at GREC 2010

**Benoit Favre**

LIUM, Université du Maine  
72000 Le Mans, France

benoit.favre@gmail.com bohnet@informatik.uni-stuttgart.de

**Bernd Bohnet**

Universität Stuttgart  
Stuttgart, Germany

## Abstract

We present the UMUS (Université du Maine/Universität Stuttgart) submission for the NEG task at GREC'10. We refined and tuned our 2009 system but we still rely on predicting generic labels and then choosing from the list of expressions that match those labels. We handled recursive expressions with care by generating specific labels for all the possible embeddings. The resulting system performs at a type accuracy of 0.84 and a string accuracy of 0.81 on the development set.

## 1 Introduction

The Named Entity Generation (NEG) task consists in choosing a referential expression (complete name, last name, pronoun, possessive pronoun, elision...) for all person entities in a text. Texts are biographies of chefs, composers and inventors from Wikipedia. For each reference, a list of expressions is given from which the system has to choose. This task is challenging because of the following aspects:

1. The data is imperfect as it is a patchwork of multiple authors' writing.
2. The problem is hard to handle with a classifier because text is predicted, not classes.
3. The problem has a complex graph structure.
4. Some decisions are recursive for embedded references, i.e. "his father".
5. Syntactic/semantic features cannot be extracted with a classical parser because the word sequence is latent.

We do not deal with all of these challenges but we try to mitigate their impact. Our system extends our approach for GREC'09 (Favre and Bohnet, 2009). We use a sequence classifier to predict generic labels for the possible expressions.

## 2 Labels for classification

Each referential expression (REFEX) is given a label consisting of sub-elements:

- The REG08\_TYPE as given in the REFEX (name, common, pronoun, empty...)
- The CASE as given in the REFEX (plain, genitive, accusative...)
- If the expression is a pronoun, then one of "he, him, his, who, whom, whose, that", after gender and number normalization.
- "self" if the expression contains "self".
- "short" if the expression is a one-word long name or common name.
- "nesting" if the expression is recursive.

For recursive expressions, a special handling is applied: All possible assignments of the embedded entities are generated with labels corresponding to the concatenation of the involved entities' labels. If the embedding is on the right (left) side of the expression, "right" ("left") is added to the label. Non-sensical labels (i.e. "he father") are not seen in the training data, and therefore not hypothesized.

## 3 Features

Each reference is characterized with the following features:

- SYNFUNC, SEMCAT, SYNCAT: syntactic function, semantic category, syntactic category, as given in REF node.
- CHANGE, CHANGE+SYNFUNC: previous reference is for a different entity, possibly with syntactic function.
- PREV\_GENDER\_NUMBER: if the reference is from a different entity, can be "same"

or “different”. The attribute is being compared is “male”, “female” or “plural”, determined by looking at the possible expressions.

- **FIRST.TIME**: denotes if it’s the first time that the entity is seen. For plural entities, the entity is considered new if at least one of the involved entities is new.
- **BEG\_PARAGRAPH**: the first entity of a paragraph.
- **{PREV,NEXT}\_PUNCT**: the punctuation immediately before (after) the entity. Can be “sentence” if the punctuation is one of “?!”, “comma” for “;”, “parenthesis” for “()[]” and “quote”.
- **{PREV,NEXT}\_SENT**: whether or not a sentence boundary occurs after (before) the previous (next) reference.
- **{PREV,NEXT}\_WORD\_{1,2}GRAM**: corresponding word n-gram. Words are extracted up to the previous/next reference or the start/end of a sentence, with parenthesized content removed. Words are lower-cased tokens made of letters and numbers.
- **{PREV,NEXT}\_TAG**: most likely part-of-speech tag for the previous/next word, skipping adverbs.
- **{PREV,NEXT}\_BE**: any form of the verb “to be” is used after (before) the previous (next) reference.
- **EMBEDS\_PREV**: the entity being embedded was referred to just before.
- **EMBEDS\_ALL\_KNOWN**: all the entities being embedded have been seen before.

#### 4 Sequence classifier

We rely on Conditional Random Fields<sup>1</sup> (Lafferty et al., 2001) for predicting one label (as defined previously) per reference. We lay the problem as one sequence of decisions per entity to prevent, for instance, the use of the same name twice in a row. Last year, we generated one sequence per document with all entities, but it was less intuitive. To the features extracted for each reference, we add the features of the previous and next reference, according to label unigrams and label bigrams. The  $c$  hyperparameter and the frequency cutoff of the classifier are optimized on the dev set. Note that

<sup>1</sup>CRF++, <http://crfpp.sourceforge.net>

for processing the test set, we added the development data to the training set.

#### 5 Text generation

For each reference, the given expressions are ranked by classifier-estimated posterior probability and the best one is used for output. In case multiple expressions have the same labeling (and the same score), we use the longest one and iterate through the list for each subsequent use (useful for repeated common names). If an expression is more than 4 words, it’s flagged for not being used a second time (only ad-hoc rule in the system).

#### 6 Results

Evaluation scores for the output are presented in Table 1. The source code of our systems is made available to the community at <http://code.google.com/p/icsicrf-grecneg>.

Sys.	T.acc	Prec.	Rec.	S.acc	Bleu	Nist
Old	0.826	0.830	0.830	0.786	0.811	5.758
New	<b>0.844</b>	0.829	0.816	<b>0.813</b>	0.817	6.021

Table 1: Results on the dev set comparing our system from last year (old) to the refined one (new), according to REG08\_TYPE accuracy (T.acc), precision and recall, String accuracy (S.acc), BLEU1 and NIST.

About 50% of the errors are caused by the selection of pronouns instead of a name. The selection of the pronoun or name seems to depend on the writing style since a few authors prefer nearly always the name. The misuse of names instead of pronouns is second most error with about 15%. The complex structured named entities are responsible for about 9% of the errors. The selection of the right name such as given name, family name or both seems to be more difficult. The next frequent errors are confusions between pronouns, elisions, common names, and names.

#### References

- Benoit Favre and Bernd Bonhet. 2009. ICSI-CRF: The Generation of References to the Main Subject and Named Entities Using Conditional Random Fields. In *ACL-IJCNLP*.
- J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Machine Learning*, pages 282–289.

# UDeL: Refining a Method of Named Entity Generation

Charles F. Greenbacker, Nicole L. Sparks, Kathleen F. McCoy, and Che-Yu Kuo

Department of Computer and Information Sciences

University of Delaware

Newark, Delaware, USA

[charlieg|sparks|mccoy|kuo]@cis.udel.edu

## Abstract

This report describes the methods and results of a system developed for the GREC Named Entity Challenge 2010. We detail the refinements made to our 2009 submission and present the output of the self-evaluation on the development data set.

## 1 Introduction

The GREC Named Entity Challenge 2010 (NEG) is an NLG shared task whereby submitted systems must select a referring expression from a list of options for each mention of each person in a text. The corpus is a collection of 2,000 introductory sections from Wikipedia articles about individual people in which all mentions of person entities have been annotated. An in-depth description of the task, along with the evaluation results from the previous year, is provided by Belz et al. (2009).

Our 2009 submission (Greenbacker and McCoy, 2009a) was an extension of the system we developed for the GREC Main Subject Reference Generation Challenge (MSR) (Greenbacker and McCoy, 2009b). Although our system performed reasonably-well in predicting REG08-Type in the NEG task, our string accuracy scores were disappointingly-low, especially when compared to the other competing systems and our own performance in the MSR task. As suggested by the evaluators (Belz et al., 2009), this was due in large part to our reliance on the list of REs being in a particular order, which had changed for the NEG task.

## 2 Method

The first improvement we made to our existing methods related to the manner by which we selected the specific RE to employ. In 2009, we trained a series of decision trees to predict REG08-Type based on our psycholinguistically-inspired

feature set (described in (Greenbacker and McCoy, 2009c)), and then simply chose the first option in the list of REs matching the predicted type. For 2010, we incorporated the case of each RE into our target attribute so that the decision tree classifier would predict both the type and case for the given reference. Then, we applied a series of rules governing the length of initial and subsequent REs involving a person's name (following Nenkova and McKeown (2003)), as well as 'back-offs' if the predicted type or case were not available.

Another improvement we made involved our method of determining whether the use of a pronoun would introduce ambiguity in a given context. Previously, we searched for references to other people entities since the most recent mention of the entity at hand, and if any were found, we assumed these would cause the use of a pronoun to be ambiguous. However, this failed to account for the fact that personal pronouns in English are gender-specific (ie. the mention of a male individual would not make the use of "she" ambiguous). So, we refined this by determining the gender of each named entity (by seeing which personal pronouns were associated with it in the list of REs), and only noting ambiguity when the current entity and candidate interfering antecedent were of the same gender.

Other small changes from 2009 include an expanded abbreviation set in the sentence segmenter, separate decision trees for the main subject and other entities, and fixing how we handled embedded REF elements with unspecified mention IDs.

## 3 Results

Scores for REG08-Type precision & recall, string accuracy, and string-edit distance are presented in Figure 1. These were computed on the entire development set, as well as the three subsets, using the `geval.pl` self-evaluation tool provided in the

NEG participants' pack.

While we were able to achieve an improvement of nearly 50% over our 2009 scores in string accuracy, we saw less than a 1% gain in overall REG08-Type performance.

Metric	Score
Type Precision/Recall	0.757995735607676
String Accuracy	0.650496141124587
Mean Edit Distance	0.875413450937156
Normalized Distance	0.319266300067796

(a) Scores on the entire development set.

Metric	Score
Type Precision/Recall	0.735294117647059
String Accuracy	0.623287671232877
Mean Edit Distance	0.839041095890411
Normalized Distance	0.345490867579909

(b) Scores on the 'Chefs' subset.

Metric	Score
Type Precision/Recall	0.790769230769231
String Accuracy	0.683544303797468
Mean Edit Distance	0.882911392405063
Normalized Distance	0.279837251356239

(c) Scores on the 'Composers' subset.

Metric	Score
Type Precision/Recall	0.745928338762215
String Accuracy	0.642140468227425
Mean Edit Distance	0.903010033444816
Normalized Distance	0.335326519731057

(d) Scores on the 'Inventors' subset.

Figure 1: Scores on the development set obtained via the `geval.pl` self-evaluation tool. REG08-Type precision and recall were equal in all four sets.

## 4 Conclusions

The fact that our string accuracy scores improved over our 2009 submission far more than REG08-Type prediction is hardly surprising. Our efforts during this iteration of the NEG task were primarily focused on enhancing our methods of choosing the best RE once the reference type was selected.

We remain several points below the best-performing team from 2009 (ICSI-Berkeley), possibly due to the inclusion of additional items in their feature set, or the use of Conditional Random Fields as their learning technique (Favre and Bohnet, 2009).

## 5 Future Work

Moving forward, we hope to expand our feature set by including the morphology of words immediately surrounding the reference, as well as a more extensive reference history, as suggested by (Favre and Bohnet, 2009). We suspect that these features may play a significant role in determining the type of referenced used, the prediction of which acts as a 'bottleneck' in generating exact REs.

We would also like to compare the efficacy of several different machine learning techniques as applied to our feature set and the NEG task.

## References

- Anja Belz, Eric Kow, and Jette Viethen. 2009. The GREC named entity generation challenge 2009: Overview and evaluation results. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 88–98, Suntec, Singapore, August. Association for Computational Linguistics.
- Benoit Favre and Bernd Bohnet. 2009. ICSI-CRF: The generation of references to the main subject and named entities using conditional random fields. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 99–100, Suntec, Singapore, August. Association for Computational Linguistics.
- Charles Greenbacker and Kathleen McCoy. 2009a. UDel: Extending reference generation to multiple entities. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 105–106, Suntec, Singapore, August. Association for Computational Linguistics.
- Charles Greenbacker and Kathleen McCoy. 2009b. UDel: Generating referring expressions guided by psycholinguistic findings. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 101–102, Suntec, Singapore, August. Association for Computational Linguistics.
- Charles F. Greenbacker and Kathleen F. McCoy. 2009c. Feature selection for reference generation as informed by psycholinguistic research. In *Proceedings of the CogSci 2009 Workshop on Production of Referring Expressions (PRE-Cogsci 2009)*, Amsterdam, July.
- Ani Nenkova and Kathleen McKeown. 2003. Improving the coherence of multi-document summaries: a corpus study for modeling the syntactic realization of entities. Technical Report CUCS-001-03, Columbia University, Computer Science Department.



# UDel: Named Entity Recognition and Reference Regeneration from Surface Text

Nicole L. Sparks, Charles F. Greenbacker, Kathleen F. McCoy, and Che-Yu Kuo

Department of Computer and Information Sciences

University of Delaware

Newark, Delaware, USA

[sparks|charlieg|mccoy|kuo]@cis.udel.edu

## Abstract

This report describes the methods and results of a system developed for the GREC Named Entity Recognition and GREC Named Entity Regeneration Challenges 2010. We explain our process of automatically annotating surface text, as well as how we use this output to select improved referring expressions for named entities.

## 1 Introduction

Generation of References in Context (GREC) is a set of shared task challenges in NLG involving a corpus of introductory sentences from Wikipedia articles. The Named Entity Recognition (GREC-NER) task requires participants to recognize all mentions of people in a document and indicate which mentions corefer. In the Named Entity Regeneration (GREC-Full) task, submitted systems attempt to improve the clarity and fluency of a text by generating improved referring expressions (REs) for all references to people. Participants are encouraged to use the output from GREC-NER as input for the GREC-Full task. To provide ample opportunities for improvement, a certain portion of REs in the corpus have been replaced by more-specified named references. Ideally, the GREC-Full output will be more fluent and have greater referential clarity than the GREC-NER input.

## 2 Method

The first step in our process to complete the GREC-NER task is to prepare the corpus for input into the parser by stripping all XML tags and segmenting the text into sentences. This is accomplished with a simple script based on common abbreviations and sentence-final punctuation.

Next, the files are run through the Stanford Parser (The Stanford Natural Language Processing Group, 2010), providing a typed dependency

representation of the input text from which we extract the syntactic functions (SYNFUNC) of, and relationships between, words in the sentence.

The unmarked segmented text is also used as input for the Stanford Named Entity Recognizer (The Stanford Natural Language Processing Group, 2009). We eliminate named entity tags for locations and organizations, leaving only person entities behind. We find the pronouns and common nouns (e.g. “grandmother”) referring to person entities that the NER tool does not tag. We also identify the REG08-Type and case for each RE. Entities found by the NER tool are marked as names, and the additional REs we identified are marked as either pronouns or common nouns. Case values are determined by analyzing the assigned type and any type dependency representation (provided by the parser) involving the entity. At this stage we also note the gender of each pronoun and common noun, the plurality of each reference, and begin to deal with embedded entities.

The next step identifies which tagged mentions corefer. We implemented a coreference resolution tool using a shallow rule-based approach inspired by Lappin and Leass (1994) and Bontcheva et al. (2002). Each mention is compared to all previously-seen entities on the basis of case, gender, SYNFUNC, plurality, and type. Each entity is then evaluated in order of appearance and compared to all previous entities starting with the most recent and working back to the first in the text. We apply rules to each of these pairs based on the REG08-Type attribute of the current entity. Names and common nouns are analyzed using string and word token matching. We collected extensive, cross-cultural lists of male and female first names to help identify the gender of named entities, which we use together with SYNFUNC values for pronoun resolution. Separate rules govern gender-neutral pronouns such as “who.” By the end of this stage, we have all of the resources

Corpus	MUC-6			CEAF			B-CUBED		
	F	prec.	recall	F	prec.	recall	F	prec.	recall
Entire Set	71.984	69.657	74.471	68.893	68.893	68.893	72.882	74.309	71.509
Chefs	71.094	65.942	77.119	65.722	65.722	65.722	71.245	69.352	73.244
Composers	68.866	66.800	71.064	68.672	68.672	68.672	71.929	73.490	70.433
Inventors	76.170	77.155	75.210	72.650	72.650	72.650	75.443	80.721	70.812

Table 1: Self-evaluation scores for GREC-NER.

necessary to complete the GREC-NER task.

As a post-processing step, we remove all extra (non-GREC) tags used in previous steps, re-order the remaining attributes in the proper sequence, add the list of REs (ALT-REFEX), and write the final output following the GREC format. At this point, the GREC-NER task is concluded and its output is used as input for the GREC-Full task.

To improve the fluency and clarity of the text by regenerating the referring expressions, we rely on the system we developed for the GREC Named Entity Challenge 2010 (NEG), a refined version of our 2009 submission (Greenbacker and McCoy, 2009a). This system trains decision trees on a psycholinguistically-inspired feature set (described by Greenbacker and McCoy (2009b)) extracted from a training corpus. It predicts the most appropriate reference type and case for the given context, and selects the best match from the list of available REs. For the GREC-Full task, however, instead of using the files annotated by the GREC organizers as input, we use the files we annotated automatically in the GREC-NER task. By keeping the GREC-NER output in the GREC format, our NEG system was able to successfully run unmodified and generate our output for GREC-Full.

### 3 Results

Scores calculated by the GREC self-evaluation tools are provided in Table 1 for GREC-NER and in Table 2 for GREC-Full.

Corpus	NIST	BLEU-4
Entire Set	8.1500	0.7953
Chefs	7.5937	0.7895
Composers	7.5381	0.8026
Inventors	7.5722	0.7936

Table 2: Self-evaluation scores for GREC-Full.

### 4 Conclusions

Until we compare our results with others teams or an oracle, it is difficult to gauge our performance. However, at this first iteration of these tasks, we're pleased just to have end-to-end RE regeneration working to completion with meaningful output.

### 5 Future Work

Future improvements to our coreference resolution approach involve analyzing adjacent text, utilizing more of the parser output, and applying machine learning to our GREC-NER methods.

### References

- Kalina Bontcheva, Marin Dimitrov, Diana Maynard, Valentin Tablan, and Hamish Cunningham. 2002. Shallow Methods for Named Entity Coreference Resolution. In *Chaînes de références et résolveurs d'anaphores, workshop TALN 2002*, Nancy, France.
- Charles Greenbacker and Kathleen McCoy. 2009a. UDel: Extending reference generation to multiple entities. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UC-NLG+Sum 2009)*, pages 105–106, Suntec, Singapore, August. Association for Computational Linguistics.
- Charles F. Greenbacker and Kathleen F. McCoy. 2009b. Feature selection for reference generation as informed by psycholinguistic research. In *Proceedings of the CogSci 2009 Workshop on Production of Referring Expressions (PRE-Cogsci 2009)*, Amsterdam, July.
- Shalom Lappin and Herbert J. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–561.
- The Stanford Natural Language Processing Group. 2009. Stanford Named Entity Recognizer. <http://nlp.stanford.edu/software/CRF-NER.shtml>.
- The Stanford Natural Language Processing Group. 2010. The Stanford Parser: A statistical parser. <http://nlp.stanford.edu/software/lex-parser.shtml>.

# Report on the Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2)

**Alexander Koller**  
Saarland University

koller@mmci.uni-saarland.de

**Kristina Striegnitz**  
Union College

striegnk@union.edu

**Andrew Gargett**  
Saarland University

gargett@mmci.uni-saarland.de

**Donna Byron**  
Northeastern University

dbyron@ccs.neu.edu

**Justine Cassell**  
Northwestern University

justine@northwestern.edu

**Robert Dale**  
Macquarie University

Robert.Dale@mq.edu.au

**Johanna Moore**  
University of Edinburgh

J.Moore@ed.ac.uk

**Jon Oberlander**  
University of Edinburgh

J.Oberlander@ed.ac.uk

## Abstract

We describe the second installment of the Challenge on Generating Instructions in Virtual Environments (GIVE-2), a shared task for the NLG community which took place in 2009-10. We evaluated seven NLG systems by connecting them to 1825 users over the Internet, and report the results of this evaluation in terms of objective and subjective measures.

## 1 Introduction

This paper reports on the methodology and results of the Second Challenge on Generating Instructions in Virtual Environments (GIVE-2), which we ran from August 2009 to May 2010. GIVE is a shared task for the NLG community which we ran for the first time in 2008-09 (Koller et al., 2010). An NLG system in this task must generate instructions which guide a human user in solving a treasure-hunt task in a virtual 3D world, in real time. For the evaluation, we connect these NLG systems to users over the Internet, which makes it possible to collect large amounts of evaluation data cheaply.

While the GIVE-1 challenge was a success, in that it evaluated five NLG systems on data from 1143 game runs in the virtual environments, it was limited in that users could only move and turn in discrete steps in the virtual environments. This made the NLG task easier than intended; one of the best-performing GIVE-1 systems generated instructions of the form “move three steps forward”. The primary change in GIVE-2 compared to GIVE-1 is that users could now move and turn freely, which makes expressions like “three steps” meaningless, and makes it hard to predict the precise effect of instructing a user to “turn left”.

We evaluated seven NLG systems from six institutions in GIVE-2 over a period of three months

from February to May 2010. During this time, we collected 1825 games that were played by users from 39 countries, which is an increase of over 50% over the data we collected in GIVE-1. We evaluated each system both on objective measures (success rate, completion time, etc.) and subjective measures which were collected by asking the users to fill in a questionnaire. We completely revised the questionnaire for the second challenge, which now consists of relatively fine-grained questions that can be combined into more high-level groups for reporting. We also introduced several new objective measures, including the point in the game in which users lost or cancelled, and an experimental “back-to-base” task intended to measure how much users learned about the virtual world while interacting with the NLG system.

**Plan of the paper.** The paper is structured as follows. In Section 2, we describe and motivate the GIVE-2 Challenge. In section 3, we describe the evaluation method and infrastructure. Section 4 reports on the evaluation results. Finally, we conclude and discuss future work in Section 5.

## 2 The GIVE Challenge

GIVE-2 is the second installment of the GIVE Challenge (“Generating Instructions in Virtual Environments”), which we ran for the first time in 2008-09. In the GIVE scenario, subjects try to solve a treasure hunt in a virtual 3D world that they have not seen before. The computer has a complete symbolic representation of the virtual world. The challenge for the NLG system is to generate, in real time, natural-language instructions that will guide the users to the successful completion of their task.

Users participating in the GIVE evaluation start the 3D game from our website at [www.give-challenge.org](http://www.give-challenge.org). They then see a 3D



Figure 1: What the user sees when playing with the GIVE Challenge.

game window as in Fig. 1, which displays instructions and allows them to move around in the world and manipulate objects. The first room is a tutorial room where users learn how to interact with the system; they then enter one of three evaluation worlds, where instructions for solving the treasure hunt are generated by an NLG system. Users can either finish a game successfully, lose it by triggering an alarm, or cancel the game. This result is stored in a database for later analysis, along with a complete log of the game.

In each game world we used in GIVE-2, players must pick up a trophy, which is in a wall safe behind a picture. In order to access the trophy, they must first push a button to move the picture to the side, and then push another sequence of buttons to open the safe. One floor tile is alarmed, and players lose the game if they step on this tile without deactivating the alarm first. There are also a number of distractor buttons which either do nothing when pressed or set off an alarm. These distractor buttons are intended to make the game harder and, more importantly, to require appropriate reference to objects in the game world. Finally, game worlds contained a number of objects such as chairs and flowers that did not bear on the task, but were available for use as landmarks in spatial descriptions generated by the NLG systems.

The crucial difference between this task and the (very similar) GIVE-1 task was that in GIVE-2, players could move and turn freely in the virtual world. This is in contrast to GIVE-1, where players could only turn by 90 degree increments, and jump forward and backward by discrete steps. This feature of the way the game controls were set

up made it possible for some systems to do very well in GIVE-1 with only minimal intelligence, using exclusively instructions such as “turn right” and “move three steps forward”. Such instructions are unrealistic – they could not be carried over to instruction-giving in the real world –, and our aim was to make GIVE harder for systems that relied on them.

### 3 Method

Following the approach from the GIVE-1 Challenge (Koller et al., 2010), we connected the NLG systems to users over the Internet. In each game run, one user and one NLG system were paired up, with the system trying to guide the user to success in a specific game world.

#### 3.1 Software infrastructure

We adapted the GIVE-1 software to the GIVE-2 setting. The GIVE software infrastructure (Koller et al., 2009a) consists of three different modules: The *client*, which is the program which the user runs on their machine to interact with the virtual world (see Fig. 1); a collection of *NLG servers*, which generate instructions in real-time and send them to the client; and a *matchmaker*, which chooses a random NLG server and virtual world for each incoming connection from a client and stores the game results in a database.

The most visible change compared to GIVE-1 was to modify the client so it permitted free movement in the virtual world. This change further necessitated a number of modifications to the internal representation of the world. To support the development of virtual worlds for GIVE, we changed the file format for world descriptions to be much more readable, and provided an automatic tool for displaying virtual worlds graphically (see the screenshots in Fig. 2).

#### 3.2 Recruiting subjects

Participants were recruited using email distribution lists and press releases posted on the Internet and in traditional newspapers. We further advertised GIVE at the Cebit computer expo as part of the Saarland University booth. Recruiting anonymous experimental subjects over the Internet carries known risks (Gosling et al., 2004), but we showed in GIVE-1 that the results obtained for the GIVE Challenge are comparable and more informative than those obtained from a laboratory-

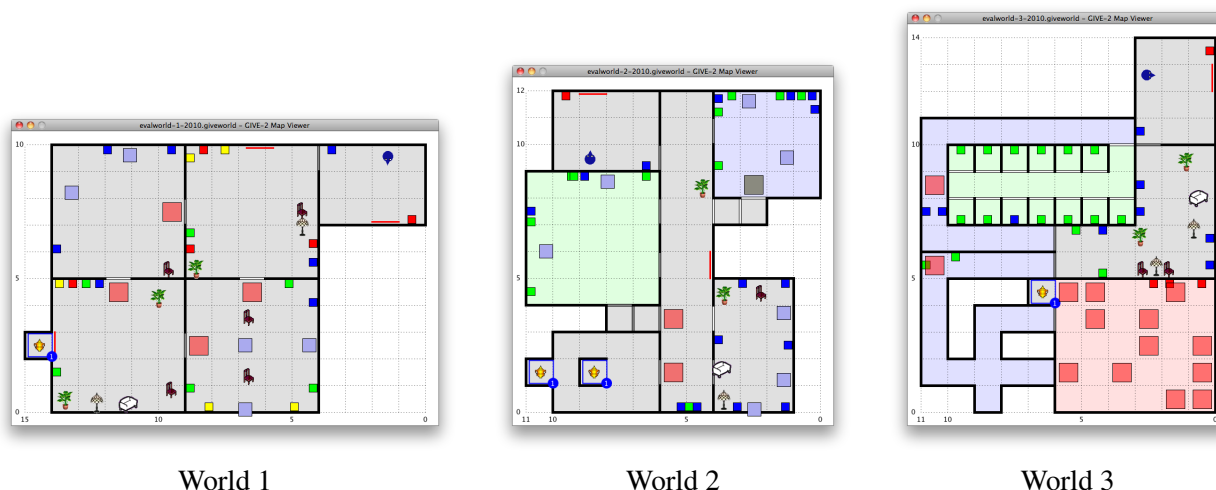


Figure 2: The three GIVE-2 evaluation worlds.

based experiment (Koller et al., 2009b).

We also tried to leverage social networks for recruiting participants by implementing and advertising a Facebook application. Because of a software bug, only about 50 participants could be recruited in this way. Thus tapping the true potential of social networks for recruiting participants remains a task for the next installment of GIVE.

### 3.3 Evaluation worlds

Fig. 2 shows the three virtual worlds we used in the GIVE-2 evaluation. Overall, the worlds were more difficult than the worlds used in GIVE-1, where some NLG-systems had success rates around 80% in some of the worlds. As for GIVE-1, the three worlds were designed to pose different challenges to the NLG systems. World 1 was intended to be more similar to the development world and last year's worlds. It did have rooms with more than one button of the same color, however, these buttons were not located close together. World 2 contained several situations which required more sophisticated referring expressions, such as rooms with several buttons of the same color (some of them close together) and a grid of buttons. Finally, World 3 was designed to exercise the systems' navigation instructions: one room contained a "maze" of alarm tiles, and another room two long rows of buttons hidden in "booths" so that they were not all visible at the same time.

### 3.4 Timeline

After the GIVE-2 Challenge was publicized in June 2009, fifteen researchers and research teams declared their interest in participating. We dis-

tributed a first version of the software to these teams in August 2009. In the end, six teams submitted NLG systems (two more than in GIVE-1); one team submitted two independent NLG systems, bringing the total number of NLG systems up to seven (two more than in GIVE-1). These were connected to a central matchmaker that ran for a bit under three months, from 23 February to 17 May 2010.

### 3.5 NLG systems

Seven NLG systems were evaluated in GIVE-2:

- one system from the Dublin Institute of Technology ("D" in the discussion below);
- one system from Trinity College Dublin ("T");
- one system from the Universidad Complutense de Madrid ("M");
- one system from the University of Heidelberg ("H");
- one system from Saarland University ("S");
- and two systems from INRIA Grand-Est in Nancy ("NA" and "NM").

Detailed descriptions of these systems as well as each team's own analysis of the evaluation results can be found at <http://www.give-challenge.org/research>.

## 4 Results

We now report the results of GIVE-2. We start with some basic demographics; then we discuss objective and subjective evaluation measures. The data for the objective measures are extracted from

the logs of the interactions; whereas the data for the subjective measures are obtained from a questionnaire which asked subjects to rate various aspects of the NLG system they interacted with.

Notice that some of our evaluation measures are in tension with each other: For instance, a system which gives very low-level instructions may allow the user to complete the task more quickly (there is less chance of user errors), but it will require more instructions than a system that aggregates these. This is intentional, and emphasizes our desire to make GIVE a friendly comparative challenge rather than a competition with a clear winner.

#### 4.1 Demographics

Over the course of three months, we collected 1825 valid games. This is an increase of almost 60% over the number of valid games we collected in GIVE-1. A game counted as valid if the game client did not crash, the game was not marked as a test game by the developers, and the player completed the tutorial.

Of these games, 79.0% were played by males and 9.6% by females; a further 11.4% did not specify their gender. These numbers are comparable to GIVE-1. About 42% of users connected from an IP address in Germany; 12% from the US, 8% from France, 6% from Great Britain, and the rest from 35 further countries. About 91% of the participants who answered the question self-rated their English language proficiency as “good” or better. About 65% of users connected from various versions of Windows, the rest were split about evenly between Linux and MacOS.

#### 4.2 Objective measures

The objective measures are summarized in Fig. 3. In addition to calculating the percentage of games users completed successfully when being guided by the different systems, we measured the time until task completion, the distance traveled until task completion, and the number of actions (such as pushing a button to open a door) executed. Furthermore, we counted how many instructions users received from each system, and how many words these instructions contained on average. All objective measures were collected completely unobtrusively, without requiring any action on the user’s part. To ensure comparability, we only counted successfully completed games.

**task success:** Did the player get the trophy?

**duration:** Time in seconds from the end of the tutorial until the retrieval of the trophy.

**distance:** Distance traveled (measured in distance units of the virtual environment).

**actions:** Number of object manipulation actions.

**instructions:** Number of instructions produced by the NLG system.

**words per instruction:** Average number of words the NLG system used per instruction.

Figure 3: Objective measures.

Fig. 4 shows the results of these objective measures. Task success is reported as the percentage of successfully completed games. The other measures are reported as the mean number of seconds/distance units/actions/instructions/words per instruction, respectively. The figure also assigns systems to groups A, B, etc. for each evaluation measure. For example, users interacting with systems in group A had a higher task success rate, needed less time, etc. than users interacting with systems in group B. If two systems do *not* share the same letter, the difference between these two systems is significant with  $p < 0.05$ . Significance was tested using a  $\chi^2$ -test for task success and ANOVAs for the other objective measures. These were followed by post-hoc tests (pairwise  $\chi^2$  and Tukey) to compare the NLG systems pairwise.

In terms of task success, the systems fall pretty neatly into four groups. Note that systems D and T had very low task success rates. That means that, for these systems, the results for the other objective measures may not be reliable because they are based on just a handful of games. Another aspect in which systems clearly differed is how many words they used per instruction. Interestingly, the three systems with the best task success rates also produced the most succinct instructions. The distinctions between systems in terms of the other measures is less clear.

#### 4.3 Subjective measures

The subjective measures were obtained from responses to a questionnaire that was presented to users after each game. The questionnaire asked users to rate different statements about the NLG

	D	H	M	NA	NM	S	T
task success	9%	11%	13%	47%	30%	40%	3%
				A		A	
					B		
	C	C	C				
	D						D
duration	888	470	407	344	435	467	266
		A	A	A	A		A
		B	B		B	B	B
	C						
distance	231	164	126	162	167	150	89
		A	A	A	A	A	A
	B	B		B	B		B
actions	25	22	17	17	18	17	14
	A	A	A	A	A	A	A
instructions	349	209	463	224	244	244	78
	A	A		A	A	A	A
	B		B				
words per instruction	15	11	16	6	10	6	18
				A		A	
					B		
		C					
	D						
			E				E

Figure 4: Results for the *objective* measures.

system using a continuous slider. The slider position was translated to a number between -100 and 100. Figs. 7 and 6 show the statements that users were asked to rate as well as the results. These results are based on all games, independent of the success. We report the mean rating for each item, and, as before, systems that do not share a letter, were found to be significantly different ( $p < 0.05$ ). We used ANOVAs and post-hoc Tukey tests to test for significance. Note that some items make a positive statement about the NLG system (e.g., Q1) and some make a negative statement (e.g., Q2). For negative statements, we report the reversed scores, so that in Figs. 7 and 6 greater numbers are always better, and systems in group A are always better than systems in group B.

In addition to the items Q1–Q22, the questionnaire contained a statement about the overall instruction quality: “Overall, the system gave me good directions.” Furthermore notice that the other items fall into two categories: items that assess the quality of the instructions (Q1–Q15) and items that assess the emotional affect of the interaction (Q16–Q22). The ratings in these cate-

	D	H	M	NA	NM	S	T
overall quality question	-33	-18	-12	36	18	19	-25
				A		B	B
	C	C	C				C
quality measures (summed)	-183	-148	-18	373	239	206	-44
	B	B	B	A	A	A	B
emotional affect measures (summed)	-130	-103	-90	20	-5	0	-88
				A	A	A	A
	B		B		B	B	B
	C	C	C		C		C

Figure 5: Results for item assessing overall instruction quality and the aggregated quality and emotional affect measures.

gories can be aggregated into just two ratings by summing over them. Fig. 5 shows the results for the overall question and the aggregated ratings for quality measures and emotional affect measures. The three systems with the highest task success rate get rated highest for overall instruction quality. The aggregated quality measure also singles out the same group of three systems.

#### 4.4 Further analysis

In addition to the differences between NLG systems, some other factors also influence the outcomes of our objective and subjective measures. As in GIVE-1, we find that there is a significant difference in task success rate for different evaluation worlds and between users with different levels of English proficiency. Fig. 8 illustrates the effect of the different evaluation worlds on the task success rate for different systems, and Fig. 9 shows the effect that a player’s English skills have on the task success rate. As in GIVE-1, some systems seem to be more robust than others with respect to changes in these factors.

None of the other factors we looked at (gender, age, and computer expertise) have a significant effect on the task success rate. With a few exceptions the other objective measures were not influenced by these demographic factors either. However, we do find a significant effect of age on the time and number of actions a player needs to retrieve the trophy: younger players are faster and need fewer actions. And we find that women travel a significantly shorter distance than men on their way to the trophy. Interestingly, we do not find

D	H	M	NA	NM	S	T
Q1: The system used words and phrases that were easy to understand.						
45	26	41	62	54	58	46
			A	A	A	A
B		B		B		B
C	C	C				
Q2: I had to re-read instructions to understand what I needed to do.						
-26	-9	3	40	8	19	0
			A			
		B		B	B	B
	C	C				C
D	D					
Q3: The system gave me useful feedback about my progress.						
-17	-30	-31	9	11	-13	-27
			A	A		
B		B			B	B
C	C	C				C
Q4: I was confused about what to do next.						
-35	-27	-18	29	9	5	-31
			A			
				B	B	
C	C	C				C
Q5: I was confused about which direction to go in.						
-32	-20	-16	21	8	3	-25
			A	A		
				B	B	
C	C	C				C
Q6: I had no difficulty with identifying the objects the system described for me.						
-21	-11	-5	18	13	20	-21
			A	A	A	
		B		B		
C	C	C				C
Q7: The system gave me a lot of unnecessary information.						
-22	-9	6	15	10	10	-6
		A	A	A	A	
		B		B	B	B
	C	C				C
D	D					D
Q8: The system gave me too much information all at once.						
-28	-8	9	31	8	21	15
			A		A	A
		B		B	B	B
C	C					
Q9: The system immediately offered help when I was in trouble.						
-15	-13	-13	32	3	-5	-23
			A			
B	B	B		B	B	
C	C				C	C
Q10: The system sent instructions too late.						
15	15	9	38	39	14	8
			A	A		
B	B	B			B	B
Q11: The system's instructions were delivered too early.						
15	5	21	39	12	30	28
			A		A	A
B		B			B	B
C		C		C		C
D	D	D		D		
Q12: The system's instructions were visible long enough for me to read them.						
-67	-21	-19	6	-14	0	-18
			A		A	
				B	B	B
	C	C		C		C
D						
Q13: The system's instructions were clearly worded.						
-20	-9	1	32	23	26	6
			A	A	A	
				B	B	B
	C	C				C
D	D					
Q14: The system's instructions sounded robotic.						
16	-6	8	-4	-1	5	1
A		A	A	A	A	A
	B	B	B	B	B	B
Q15: The system's instructions were repetitive.						
-28	-26	-11	-31	-28	-26	-23
A	A	A			A	A
B	B		B	B	B	B

Figure 7: Results for the *subjective* measures assessing the *quality* of the instructions.



D	H	M	NA	NM	S	T
Q16: I really wanted to find that trophy.						
-10	-13	-9	-11	-8	-7	-12
A	A	A	A	A	A	A
Q17: I lost track of time while solving the overall task.						
-13	-18	-21	-16	-18	-11	-20
A	A	A	A	A	A	A
Q18: I enjoyed solving the overall task.						
-21	-23	-20	-8	-4	-5	-21
A		A	A	A	A	A
B	B	B	B			B
Q19: Interacting with the system was really annoying.						
-14	-20	-12	8	-2	-2	-14
			A	A	A	
B		B		B	B	B
C	C	C				C
Q20: I would recommend this game to a friend.						
-36	-39	-31	-30	-25	-24	-31
A	A	A	A	A	A	A
Q21: The system was very friendly.						
0	-1	5	30	20	19	5
			A	A	A	
		B		B	B	B
C		C			C	C
D	D	D				D
Q22: I felt I could trust the system's instructions.						
-21	-6	-3	37	23	21	-13
			A	A	A	
B	B	B				B

Figure 6: Results for the *subjective* measures assessing the *emotional affect* of the instructions.

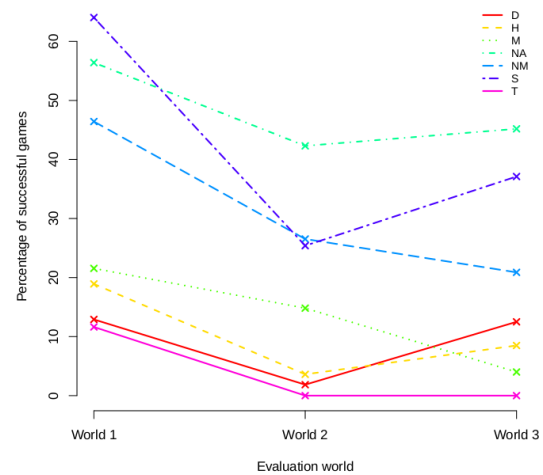


Figure 8: Effect of the evaluation worlds on the success rate of the NLG systems.

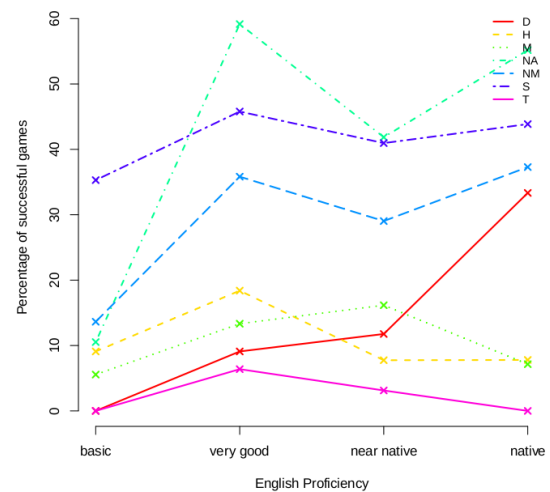


Figure 9: Effect of the players' English skills on the success rate of the NLG systems.

a significant effect of gender on the time players need to retrieve the trophy as in GIVE-1 (although the mean duration is somewhat higher for female than for male players; 481 vs. 438 seconds).

## 5 Conclusion

In this paper, we have described the setup and results of the Second GIVE Challenge. Altogether, we collected 1825 valid games for seven NLG systems over a period of three months. Given that this is a 50% increase over GIVE-1, we feel that this further justifies our basic experimental methodology. As we are writing this, we are preparing detailed results and analyses for each participating team, which we hope will help them understand and improve the performance of their systems.

The success rate is substantially worse in GIVE-2 than in GIVE-1. This is probably due to the

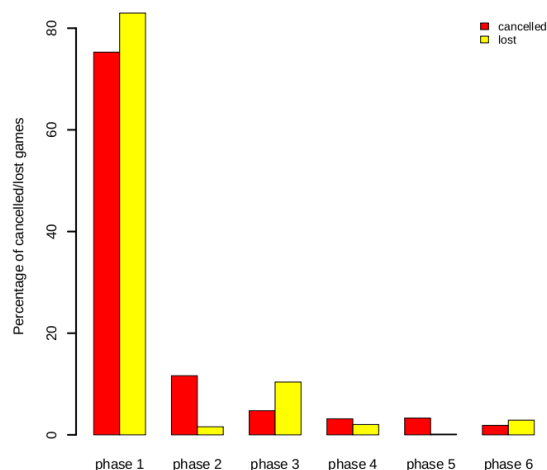


Figure 10: Points at which players lose/cancel.

harder task (free movement) explained in Section 2 and to the more complex evaluation worlds (see Section 3.3). It was our intention to make GIVE-2 more difficult, although we did not anticipate such a dramatic drop in performance. GIVE-2.5 next year will use the same task as GIVE-2 and we hope to see an increase in task success as the participating research teams learn from this year's results.

It is also noticeable that players gave mostly negative ratings in response to statements about immersion and engagement (Q16-Q20). We discussed last year how to make the task more engaging on the one hand and how to manage expectations on the other hand, but none of the suggested solutions ended up being implemented. It seems that we need to revisit this issue.

Another indication that the task may not be able to capture participants is that the vast majority of cancelled and lost games end in the very beginning. To analyze at what point players lose or give up, we divide the game into phases demarcated by manipulations of buttons that belong to the 6-button safe sequence. Fig. 10 illustrates in which phase of the game players lose or cancel.

We are currently preparing the GIVE-2.5 Challenge, which will take place in 2010-11. GIVE-2.5 will be very similar to GIVE-2, so that GIVE-2 systems will be able to participate with only minor changes. In order to support the development of GIVE-2.5 systems, we have collected a multilingual corpus of written English and German instructions in the GIVE-2 environment (Gargett et al., 2010). We expect that GIVE-3 will then extend the GIVE task substantially, perhaps in the direction of full dialogue or of multimodal interaction.

**Acknowledgments.** GIVE-2 was only possible through the support and hard work of a number of colleagues, especially Konstantina Garoufi (who handled the website and other publicity-related issues), Ielka van der Sluis (who contributed to the design of the GIVE-2 questionnaire), and several student assistants who programmed parts of the GIVE-2 system. We thank the press offices of Saarland University, the University of Edinburgh, and Macquarie University for their helpful press releases. We also thank the organizers of Generation Challenges 2010 and INLG 2010 for their support and the opportunity to present our results, and the seven participating research teams for their contributions.

## References

- Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 corpus of giving instructions in virtual environments. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, Malta.
- S. D. Gosling, S. Vazire, S. Srivastava, and O. P. John. 2004. Should we trust Web-based studies? A comparative analysis of six preconceptions about Internet questionnaires. *American Psychologist*, 59:93–104.
- A. Koller, D. Byron, J. Cassell, R. Dale, J. Moore, J. Oberlander, and K. Striegnitz. 2009a. The software architecture for the first challenge on generating instructions in virtual environments. In *Proceedings of the EACL-09 Demo Session*.
- Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Sara Dalziel-Job, Johanna Moore, and Jon Oberlander. 2009b. Validating the web-based evaluation of NLG systems. In *Proceedings of ACL-IJCNLP 2009 (Short Papers)*, Singapore.
- Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The first challenge on generating instructions in virtual environments. In E. Kraemer and M. Theune, editors, *Empirical Methods in Natural Language Generation*, volume 5790 of *LNCS*, pages 337–361. Springer.

# The First Question Generation Shared Task Evaluation Challenge

Vasile Rus<sup>1</sup>, Brendan Wyse<sup>2</sup>, Paul Pivék<sup>2</sup>, Mihai Lintean<sup>1</sup>, Svetlana Stoyanchev<sup>2</sup> and Cristian Moldovan<sup>1</sup>

<sup>1</sup>Department of Computer Science  
Institute for Intelligent Systems  
The University of Memphis  
Memphis, TN, USA  
{vrus, mclinten, cmoldovan}  
@memphis.edu

<sup>2</sup>Centre for Research in Computing  
Open University, UK  
bjwyse@gmail.com and  
{p.pivék, s.stoyanchev}  
@open.ac.uk

## Abstract

The paper briefly describes the First Shared Task Evaluation Challenge on Question Generation that took place in Spring 2010. The campaign included two tasks: Task A – Question Generation from Paragraphs and Task B – Question Generation from Sentences. An overview of each of the tasks is provided.

## 1 Introduction

Question Generation is an essential component of learning environments, help systems, information seeking systems, multi-modal conversations between virtual agents, and a myriad of other applications (Lauer, Peacock, and Graesser, 1992; Piwek et al., 2007).

Question Generation has been recently defined as the task (Rus & Graesser, 2009) of automatically generating questions from some form of input. The input could vary from information in a database to a deep semantic representation to raw text.

The first Shared Task Evaluation Challenge on Question Generation (QG-STEC) follows a long tradition of STECs in Natural Language Processing (see the annual tasks run by the Conference on Natural Language Learning - CoNLL). In particular, the idea of a QG-STEC was inspired by the recent activity in the Natural Language Generation (NLG) community to offer shared task evaluation campaigns as a potential avenue to provide a focus for research in NLG

and to increase the visibility of NLG in the wider Natural Language Processing (NLP) community (White and Dale, 2008). It should be noted that the QG is currently perceived as a discourse processing task rather than a traditional NLG task (Rus & Graesser, 2009).

Two core aspects of a question are the goal of the question and its importance. It is difficult to determine whether a particular question is good without knowing the context in which it is posed; ideally one would like to have information about what counts as important and what the goals are in the current context. This suggests that a STEC on QG should be tied to a particular application, e.g. tutoring systems. However, an application-specific STEC would limit the pool of potential participants to those interested in the target application. Therefore, the challenge was to find a framework in which the goal and importance are intrinsic to the source of questions and less tied to a particular context/application. One possibility was to have the general goal of asking questions about salient items in a source of information, e.g. core ideas in a paragraph of text. Our tasks have been defined with this concept in mind. Adopting the basic principle of application-independence has the advantage of escaping the problem of a limited pool of participants (to those interested in a particular application had that application been chosen as the target for a QG STEC).

Another decision aimed at attracting as many participants as possible and promoting a more fair comparison environment was the input for the QG tasks. Adopting a specific representation for the input would have favored some participants already familiar with such a representation. Therefore, we have adopted as a second guiding

principle for the first QG-STECS tasks: no representational commitment. That is, we wanted to have as generic an input as possible. The input to both task A and B in the first QG STEC is raw text.

The First Workshop on Question Generation ([www.questiongeneration.org](http://www.questiongeneration.org)) has identified four categories of QG tasks (Rus & Graesser, 2009): Text-to-Question, Tutorial Dialogue, Assessment, and Query-to-Question. The two tasks in the first QG STEC are part of the Text-to-Question category or part of the Text-to-text Natural Language Generation task categories (Dale & White, 2007). It is important to say that the two tasks offered in the first QG STEC were selected among 5 candidate tasks by the members of the QG community. A preference poll was conducted and the most preferred tasks, Question Generation from Paragraphs (Task A) and Question Generation from Sentences (Task B), were chosen to be offered in the first QG STEC. The other three candidate tasks were: Ranking Automatically Generated Questions (Michael Heilman and Noah Smith), Concept Identification and Ordering (Rodney Nielsen and Lee Becker), and Question Type Identification (Vasile Rus and Arthur Graesser).

There is overlap between Task A and B. This was intentional with the aim of encouraging people preferring one task to participate in the other. The overlap consists of the specific questions in Task A which are more or less similar with the type of questions targeted by Task B.

Overall, we had 1 submission for Task A and 4 submissions for Task B. We also had an additional submission on development data for Task A.

## 2 TASK A: Question Generation from Paragraphs

### 2.1 Task Definition

The Question Generation from Paragraphs (QGP) task challenges participants to generate a list of 6 questions from a given input paragraph. The six questions should be at three scope levels: 1 x broad (entire input paragraph), 2 x medium (multiple sentences), and 3 x specific (sentence or less). The scope is defined by the portion of the paragraph that answers the question.

The Question Generation from Paragraphs (QGP) task has been defined such that it is *application-independent*. *Application-independent* means questions will be judged based on content

analysis of the input paragraph; questions whose answers span more input text are ranked higher.

Table 1 shows an example paragraph, while in Table 2 we list six interesting, application-independent questions that could be generated. We will use the paragraph and questions to describe the judging criteria.

A set of five scores, one for each criterion (specificity, syntax, semantics, question type correctness, diversity), and a composite score will be assigned to each question. Each question at each position will be assigned a composite score ranging from 1 (first/top ranked, best) to 4 (worst rank), 1 meaning the question is at the right level of specificity given its rank (e.g. the broadest question that the whole paragraph answers will get a score of 1 if in the first position) and also it is syntactically and semantically correct as well as unique/diverse from other generated questions in the set.

Ranking of questions based on scope assures a maximum score for the six questions of 1, 2, 2, 3, 3 and 3, respectively. A top-rank score of 1 is assigned to a broad scope question that is also syntactically and semantically correct or acceptable, i.e. if it is semantically ineligible then a decision about its scope cannot be made and thus a worst-rank score of 4 is assigned. A maximum score of 2 is assigned to medium-scope questions while a maximum score of 3 is assigned to specific questions. The best configuration of scores (1, 2, 2, 3, 3, 3) would only be possible for paragraphs that could trigger the required number of questions at each scope level, which may not always be the case.

### 2.2 Data Sources and Annotation

The primary source of input paragraphs were: Wikipedia, OpenLearn, Yahoo!Answers. We collected 20 paragraphs from each of these three sources. We collected both a development data set (65 paragraphs) and a test data set (60 paragraphs). For the development data set we manually generated and scored 6 questions per paragraph for a total of  $6 \times 65 = 390$  questions.

Paragraphs were selected such that they are self-contained (no need for previous context to be interpreted, e.g. will have no unresolved pronouns) and contain around 5-7 sentences for a total of 100-200 tokens (excluding punctuation). In addition, we aimed for a diversity of topics of general interest.

We also provided discourse relations based on HILDA, a freely available automatic discourse parser (duVerle & Prendinger, 2009).

**Table 1.** Example of input paragraph (from [http://en.wikipedia.org/wiki/Abraham\\_Lincoln](http://en.wikipedia.org/wiki/Abraham_Lincoln)).

Input Paragraph
<i>Abraham Lincoln (February 12, 1809 – April 15, 1865), the 16th President of the United States, successfully led his country through its greatest internal crisis, the American Civil War, preserving the Union and ending slavery. As an outspoken opponent of the expansion of slavery in the United States, Lincoln won the Republican Party nomination in 1860 and was elected president later that year. His tenure in office was occupied primarily with the defeat of the secessionist Confederate States of America in the American Civil War. He introduced measures that resulted in the abolition of slavery, issuing his Emancipation Proclamation in 1863 and promoting the passage of the Thirteenth Amendment to the Constitution. As the civil war was drawing to a close, Lincoln became the first American president to be assassinated.</i>

**Table 2.** Examples of questions and scores for the paragraph in Table 1.

Questions	Scope
<i>Who is Abraham Lincoln?</i>	<i>General</i>
<i>What major measures did President Lincoln introduce?</i>	<i>Medium</i>
<i>How did President Lincoln die?</i>	<i>Medium</i>
<i>When was Abraham Lincoln elected president?</i>	<i>Specific</i>
<i>When was President Lincoln assassinated?</i>	<i>Specific</i>
<i>What party did Abraham Lincoln belong to?</i>	<i>Specific</i>

### 3 TASK B: Question Generation from Sentences

#### 3.1 Task Definition

Participants were given a set of inputs, with each input consisting of:

- a single sentence and
- a specific target question type (e.g., WHO?, WHY?, HOW?, WHEN?).

For each input, the task was to generate 2 questions of the specified target question type.

Input sentences, 60 in total, were selected from OpenLearn, Wikipedia and Yahoo! Answers (20 inputs from each source). Extremely short or long sentences were not included. Prior to receiving the actual test data, participants were provided with a development data set consisting of sentences from the aforementioned sources and, for one or more target question types, examples of questions. These questions were manually authored and cross-checked by the team organizing Task B.

The following example is taken from the development data set. Each instance has a unique

identifier and information on the source it was extracted from. The <text> element contains the input sentence and the <question> elements contain possible questions. The <question> element has the type attribute for specification of the target question type.

```
<instance id="3">
  <id>OpenLearn</id>
  <source>A103_5</source>
  <text>
    The poet Rudyard Kipling lost his only son in the trenches in 1915.
  </text>
  <question type="who">
    Who lost his only son in the trenches in 1915?
  </question>
  <question type="when">
    When did Rudyard Kipling lose his son?
  </question>
  <question type="how many">
    How many sons did Rudyard Kipling have?
  </question>
</instance>
```

Note that input sentences were provided as raw text. Annotations were not provided. There are a variety of NLP open-source tools available to potential participants and the choice of tools and how these tools are used was considered a fundamental part of the challenge.

This task was restricted to the following question types: WHO, WHERE, WHEN, WHICH, WHAT, WHY, HOW MANY/LONG, YES/NO. Participants were provided with this list and definitions of each of the items in it.

### 3.2 Evaluation criteria for System Outputs and Human Judges

The evaluation criteria fulfilled two roles. Firstly, they were provided to the participants as a specification of the kind of questions that their systems should aim to generate. Secondly, they also played the role of guidelines for the judges of system outputs in the evaluation exercise.

For this task, five criteria were identified: relevance, question type, syntactic correctness and fluency, ambiguity, and variety. All criteria are associated with a scale from 1 to N (where N is 2, 3 or 4), with 1 being the best score and N the worst score.

The procedure for applying these criteria is as follows:

- Each of the criteria is applied *independently* of the other criteria to each of the generated questions (except for the stipulation provided below).

We need some specific stipulations for cases where no question is returned in response to an input. For each target question type, two questions are expected. Consequently, we have the following two possibilities regarding missing questions:

- *No question is returned for a particular target question type:* for each of the missing questions, the worst score is recorded for all criteria.
- *Only one question is returned:* For the missing question, the worst score is assigned on all criteria. The question that is present is scored following the criteria, with the exception of the VARIETY criterion for which the lowest possible score is assigned.

We compute the overall score on a specific criterion. We can also compute a score which aggregates the overall scores for the criteria.

## 4 Conclusions

The submissions to the first QG STEC are now being evaluated using peer-review mechanism in which participants blindly evaluate their peers questions. At least two reviews per submissions are performed with the results to be made public at the 3<sup>rd</sup> Workshop on Question Generation that will take place in June 2010.

### Acknowledgments

We are grateful to a number of people who contributed to the success of the First Shared Task Evaluation Challenge on Question Generation: Rodney Nielsen, Amanda Stent, Arthur Graesser, Jose Otero, and James Lester. Also, we would like to thank the National Science Foundation who partially supported this work through grants RI-0836259 and RI-0938239 (awarded to Vasile Rus) and the Engineering and Physical Sciences Research Council who partially supported the effort on Task B through grant EP/G020981/1 (awarded to Paul Piwek). The views expressed in this paper are solely the authors'.

### References

- Thomas W. Lauer, Eileen Peacock, and Arthur C. Graesser. 1992. (Eds.). *Questions and information systems*. Hillsdale, NJ: Erlbaum.
- Vasile Rus and Arthur C. Graesser. 2009. *Workshop Report: The Question Generation Task and Evaluation Challenge*. Institute for Intelligent Systems, Memphis, TN, ISBN: 978-0-615-27428-7.
- Paul Piwek, Hugo Hernault, Helmut Prendinger, and Mitsuru Ishizuka. 2007. T2D: Generating Dialogues between Virtual Agents Automatically from Text. In *Intelligent Virtual Agents: Proceedings of IVA07, LNAI 4722*, September 17-19, 2007, Paris, France, (Springer-Verlag, Berlin Heidelberg) pp.161-174
- Robert Dale and Michael White. 2007. (Eds.). *Position Papers of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.
- David duVerle and Helmut Prendinger. 2009. A novel discourse parser based on Support Vector Machines. *Proc 47th Annual Meeting of the Association for Computational Linguistics and the 4th Int'l Joint Conf on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP'09)*, Singapore, Aug 2009 (ACL and AFNLP), pp 665-673.

# Generation under Uncertainty

**Oliver Lemon**

Heriot-Watt University  
Edinburgh, United Kingdom  
o.lemon@hw.ac.uk

**Srini Janarthanam**

Edinburgh University  
Edinburgh, United Kingdom  
s.janarthanam@ed.ac.uk

**Verena Rieser**

Edinburgh University  
Edinburgh, United Kingdom  
vrieser@inf.ed.ac.uk

## Abstract

We invite the research community to consider challenges for NLG which arise from uncertainty. NLG systems should be able to adapt to their audience and the generation environment in general, but often the important features for adaptation are not known precisely. We explore generation challenges which could employ simulated environments to study NLG which is adaptive under uncertainty, and suggest possible metrics for such tasks. It would be particularly interesting to explore how different planning approaches to NLG perform in challenges involving uncertainty in the generation environment.

## 1 Introduction

We would like to highlight the design of NLG systems for environments where there may be incomplete or faulty information, where actions may not always have the same results, and where there may be tradeoffs between the different possible outcomes of actions and plans.

There are various sources of uncertainty in systems which employ NLG techniques, for example:

- the current state of the user / audience (e.g. their knowledge, preferred vocabulary, goals, preferences....),
- the likely user reaction to the generated output,
- the behaviour of related components (e.g. a surface realiser, or TTS module),
- noise in the environment (for spoken output),
- ambiguity of the generated output.

The problem here is to generate output that takes these types of uncertainty into account appropriately. For example, you may need to choose a referring expression for a user, even though you are not sure whether they are an expert or novice in the domain. In addition, the next time you speak to that user, you need to adapt to new information you have gained about them (Janarthanam and Lemon, 2010). The issue of uncertainty for referring expression generation has been discussed before by (Reiter, 1991; Horacek, 2005).

Another example is in planning an Information Presentation for a user, when you cannot know with certainty how they will respond to it (Rieser and Lemon, 2009; Rieser et al., 2010). In the worst case, you may even be uncertain about the user's goals or information needs (as in "POMDP" approaches to dialogue management (Young et al., 2009; Henderson and Lemon, 2008a)), but you still need to generate output for them in an appropriate way.

In particular, in interactive applications of NLG:

- each NLG action *changes* the environment state or context,
- the effect of each NLG action is *uncertain*.

Several recent approaches describe NLG tasks as different kinds of planning, e.g. (Koller and Petrick, 2008; Rieser et al., 2010; Janarthanam and Lemon, 2010), or as contextual decision making according to a cost function (van Deemter, 2009). It would be very interesting to explore how different approaches perform in NLG problems where different types of uncertainty are present in the generation environment.

In the following we discuss possible generation challenges arising from such considerations, which we hope will lead to work on an agreed shared challenge in this research community. In section 2 we briefly review recent work showing

that simulated environments can be used to evaluate generation under uncertainty, and in section 3 we discuss some possible metrics for such tasks. Section 4 concludes by considering how a useful generation challenge could be constructed using similar methods.

## 2 Generation in Uncertain Simulated Environments

Finding the best (or “optimal”) way to generate under uncertainty requires exploring the possible outcomes of actions in stochastic environments. Therefore, related research on Dialogue Strategy learning has used data-driven simulated environments as a cheap and efficient way to explore uncertainty (Lemon and Pietquin, 2007). However, building good simulated environments is a challenge in its own right, as we illustrate in the following using the examples of Information Presentation and Referring Expression Generation. We also point out the additional challenges these simulations have to face when being used for NLG.

### 2.1 User Simulations for Information Presentation

User Simulations can provide a model of probable, but uncertain, user reactions to NLG actions, and we propose that they are a useful potential direction for exploring and evaluate different approaches to handling uncertainty in generation.

User Simulations are commonly used to train strategies for Dialogue Management, see for example (Young et al., 2007). A user simulation for Information Presentation is very similar, in that it is a predictive model of the most likely next user act.<sup>1</sup> However, this NLG predicted user act does not actually change the overall dialogue state (e.g. by filling slots) but it only changes the generator state. In other words, this NLG user simulation tells us what the user is most likely to do next, *if we were to stop generating now*.

In addition to the challenges of building user simulations for learning Dialogue policies, e.g. modelling, evaluation, and available data sets (Lemon and Pietquin, 2007), a crucial decision for NLG is the level of detail needed to train sensible

policies. While high-level dialogue act descriptions may be sufficient for dialogue policies, NLG decisions may require a much finer level of detail. The finer the required detail of user reactions, the more data is needed to build data-driven simulations.

For content selection in Information Presentation tasks (choosing presentation strategy and number of attributes), for example, the level of description can still be fairly abstract. We were most interested in probability distributions over the following possible user reactions:

1. *select*: the user chooses one of the presented items, e.g. “*Yes, I’ll take that one.*”. This reply type indicates that the information presentation was sufficient for the user to make a choice.
2. *addInfo*: The user provides more attributes, e.g. “*I want something cheap.*”. This reply type indicates that the user has more specific requests, which s/he wants to specify after being presented with the current information.
3. *requestMoreInfo*: The user asks for more information, e.g. “*Can you recommend me one?*”, “*What is the price range of the last item?*”. This reply type indicates that the system failed to present the information the user was looking for.
4. *askRepeat*: The user asks the system to repeat the same message again, e.g. “*Can you repeat?*”. This reply type indicates that the utterance was either too long or confusing for the user to remember, or the TTS quality was not good enough, or both.
5. *silence*: The user does not say anything. In this case it is up to the system to take initiative.
6. *hangup*: The user closes the interaction.

We have built user simulations using n-gram models of system (*s*) and user (*u*) acts, as first introduced by (Eckert et al., 1997). In order to account for data sparsity, we apply different *discounting* (“smoothing”) techniques including automatic *back-off*, using the CMU Statistical Language Modelling toolkit (Clarkson and Rosenfeld, 1997). For example we have constructed a **bi-**

<sup>1</sup>Similar to the internal user models applied in recent work on POMDP (Partially Observable Markov Decision Process) dialogue managers (Young et al., 2007; Henderson and Lemon, 2008b; Gasic et al., 2008) for estimation of user act probabilities.



**gram** model<sup>2</sup> for the users' reactions to the system's IP structure decisions ( $P(a_{u,t}|IP_{s,t})$ ), and a **tri-gram** (i.e. IP structure + attribute choice) model for predicting user reactions to the system's combined IP structure and attribute selection decisions:  $P(a_{u,t}|IP_{s,t}, attributes_{s,t})$ .

We have evaluated the performance of these models by measuring dialogue similarity to the original data, based on the Kullback-Leibler (KL) divergence, as also used by e.g. (Cuayáhuitl et al., 2005; Jung et al., 2009; Janarthnam and Lemon, 2009). We compared the raw probabilities as observed in the data with the probabilities generated by our n-gram models using different discounting techniques for each context. All the models have a small divergence from the original data (especially the bi-gram model), suggesting that they are reasonable simulations for training and testing NLG policies (Rieser et al., 2010).

## 2.2 Other Simulated Components

In some systems, NLG decisions may also depend on related components, such as the database, subsequent generation steps, or the Text-to-Speech module for spoken generation. Building simulations for these components to capture their inherent uncertainty, again, is an interesting challenge.

For example, one might want to adapt the generated output according to the predicted TTS quality. Therefore, one needs a model of the expected/predicted TTS quality for a TTS engine (Boidin et al., 2009).

Furthermore, NLG decisions might be inputs to a stochastic sentence realiser, such as SPARKY (Stent et al., 2004). However, one might not have a fully trained stochastic sentence realiser for this domain (yet). In (Rieser et al., 2010) we therefore modelled the variance as observed in the top ranking SPARKY examples.

## 2.3 Generating Referring Expressions under uncertainty

In this section, we present an example user simulation (US) model, that simulates the dialogue behaviour of users who react to referring expressions depending on their domain knowledge. These external simulation models are different from internal user models used by dialogue systems. In

<sup>2</sup>Where  $a_{u,t}$  is the predicted next user action at time  $t$ ,  $IP_{s,t}$  was the system's Information Presentation action at  $t$ , and  $attributes_{s,t}$  is the set of attributes selected by the system at  $t$ .

particular, such models must be sensitive to a system's choices of referring expressions. The simulation has a statistical distribution of in-built knowledge profiles that determines the dialogue behaviour of the user being simulated. Uncertainty arises because if the user does not know a referring expression, then he is more *likely* to request clarification. If the user is able to interpret the referring expressions and identify the references then he is more likely to follow the system's instruction. This behaviour is simulated by the action selection models described below.

The user simulation (US) receives the system action  $A_{s,t}$  and its referring expression choices  $REC_{s,t}$  at each turn. The US responds with a user action  $A_{u,t}$  ( $u$  denoting user). This can either be a clarification request (*cr*) or an instruction response (*ir*). We used two kinds of action selection models: a corpus-driven statistical model and a hand-coded rule-based model.

## 2.4 Corpus-driven action selection model

The user simulation (US) receives the system action  $A_{s,t}$  and its referring expression choices  $REC_{s,t}$  at each turn. The US responds with a user action  $A_{u,t}$  ( $u$  denoting user). This can either be a clarification request (*cr*) or an instruction response (*ir*). The US produces a clarification request *cr* based on the class of the referent  $C(R_i)$ , type of the referring expression  $T_i$ , and the current domain knowledge of the user for the referring expression  $DK_{u,t}(R_i, T_i)$ . Domain entities whose jargon expressions raised clarification requests in the corpus were listed and those that had more than the mean number of clarification requests were classified as *difficult* and others as *easy* entities (for example, "power adaptor" is *easy* - all users understood this expression, "broadband filter" is *difficult*). Clarification requests are produced using the following model.

$$P(A_{u,t} = cr(R_i, T_i) | C(R_i), T_i, DK_{u,t}(R_i, T_i)) \\ \text{where } (R_i, T_i) \in REC_{s,t}$$

One should note that the actual literal expression is not used in the transaction. Only the entity that it is referring to ( $R_i$ ) and its type ( $T_i$ ) are used. However, the above model simulates the process of interpreting and resolving the expression and identifying the domain entity of interest in the instruction. The user identification of the entity is signified when there is no clarification request produced (i.e.  $A_{u,t} = none$ ). When no clarification

request is produced, the environment action  $EA_{u,t}$  is generated using the following model.

$$P(EA_{u,t}|A_{s,t}) \text{ if } A_{u,t}! = cr(R_i, T_i)$$

Finally, the user action is an instruction response which is determined by the system action  $A_{s,t}$ . Instruction responses can be either *provide\_info*, *acknowledgement* or *other* based on the system's instruction.

$$P(A_{u,t} = ir|EA_{u,t}, A_{s,t})$$

All the above models were trained on our corpus data using *maximum likelihood estimation* and smoothed using a variant of *Witten-Bell discounting*. According to the data, clarification requests are much more likely when jargon expressions are used to refer to the referents that belong to the `difficult` class and which the user doesn't know about. When the system uses expressions that the user knows, the user generally responds to the instruction given by the system. These user simulation models have been evaluated and found to produce behaviour that is very similar to the original corpus data, using the Kullback-Leibler divergence metric (Janarthanam and Lemon, 2010).

### 3 Metrics

Here we discuss some possible evaluation metrics that will allow different approaches to NLG under uncertainty to be compared. We envisage that other metrics should be explored, in particular those measuring adaptivity of various types.

#### 3.1 Adaptive Information Presentation

Given a suitable corpus, a data-driven evaluation function can be constructed, using a stepwise linear regression, following the PARADISE framework (Walker et al., 2000).

For example, in (Rieser et al., 2010) we build a model which selects the features which significantly influenced the users' ratings for NLG strategies in a Wizard-of-Oz study. We also assign a value to the user's reactions (*valueUserReaction*), similar to optimising task success for DM (Young et al., 2007). This reflects the fact that good Information Presentation strategies should help the user to `select` an item (*valueUserReaction* = +100) or provide more constraints `addInfo` (*valueUserReaction* =  $\pm 0$ ), but the user should not do anything else (*valueUserReaction* = -100). The regression

in equation 1 ( $R^2 = .26$ ) indicates that users' ratings are influenced by higher level and lower level features: Users like to be focused on a small set of database hits (where *#DBhits* ranges over [1-100]), which will enable them to choose an item (*valueUserReaction*), while keeping the IP utterances short (where *#sentence* was in the range [2-18]):

$$\begin{aligned} \text{Reward} = & (-1.2) \times \#DBhits \\ & + (.121) \times \text{valueUserReaction} \\ & - (1.43) \times \#sentence \end{aligned} \quad (1)$$

#### 3.2 Measuring Adaptivity of Referring Expressions

We have also designed a metric for the goal of adapting referring expressions to each user's domain knowledge. We present the Adaptation Accuracy score *AA* that calculates how accurately the agent chose the expressions for each referent *r*, with respect to the user's knowledge. Appropriateness of an expression is based on the user's knowledge of the expression. So, when the user knows the jargon expression for *r*, the appropriate expression to use is jargon, and if s/he doesn't know the jargon, an descriptive expression is appropriate. Although the user's domain knowledge is dynamically changing due to learning, we base appropriateness on the initial state, because our objective is to adapt to the initial state of the user  $DK_{u,initial}$ . However, in reality, designers might want their system to account for user's changing knowledge as well. We calculate accuracy per referent  $RA_r$  as the ratio of number of appropriate expressions to the total number of instances of the referent in the dialogue. We then calculate the overall mean accuracy over all referents as shown below.

$$\begin{aligned} RA_r &= \frac{\#(\text{appropriate\_expressions}(r))}{\#(\text{instances}(r))} \\ \text{AdaptationAccuracy} AA &= \frac{1}{\#(r)} \sum_r RA_r \end{aligned}$$

### 4 Conclusion

We have invited the research community to consider challenges for NLG which arise from uncertainty. We argue that NLG systems, like dialogue managers, should be able to adapt to their audience and the generation environment. However, often the important features for adaptation are not precisely known. We then summarised 2 potential

directions for such challenges – example generation tasks which employ simulated uncertain environments to study adaptive NLG, and discussed some possible metrics for such tasks. We hope that this will lead to discussions on a shared challenge allowing comparison of different approaches to NLG with respect to how well they handle uncertainty.

## Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project [www.classic-project.org](http://www.classic-project.org)) and from the EPSRC, project no. EP/G069840/1.

## References

- Cedric Boidin, Verena Rieser, Lonneke van der Plas, Oliver Lemon, and Jonathan Chevelu. 2009. Predicting how it sounds: Re-ranking alternative inputs to TTS using latent variables (forthcoming). In *Proc. of Interspeech/ICSLP, Special Session on Machine Learning for Adaptivity in Spoken Dialogue Systems*.
- P.R. Clarkson and R. Rosenfeld. 1997. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proc. of ESCA Eurospeech*.
- Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-computer dialogue simulation using hidden markov models. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.
- W. Eckert, E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.
- M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and S. Young. 2008. Training and Evaluation of the HIS POMDP Dialogue System in Noise. In *Proc. of SIGdial Workshop on Discourse and Dialogue*.
- James Henderson and Oliver Lemon. 2008a. Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management. In *Proceedings of ACL*.
- James Henderson and Oliver Lemon. 2008b. Mixture Model POMDPs for Efficient Handling of Uncertainty in Dialogue Management. In *Proc. of ACL*.
- Helmut Horacek. 2005. Generating referential descriptions under conditions of uncertainty. In *ENLG*.
- Srinivasan Janarathanam and Oliver Lemon. 2009. A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies. In *Proc. of SIGdial*.
- Srini Janarathanam and Oliver Lemon. 2010. Learning to adapt to unknown users: Referring expression generation in spoken dialogue systems. In *Proceedings of ACL*. (to appear).
- Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Minwoo Jeong, and Gary Geunbae Lee. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer, Speech & Language*, 23:479–509.
- Alexander Koller and Ronald Petrick. 2008. Experiences with planning for natural language generation. In *ICAPS*.
- Oliver Lemon and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *Inter-speech*.
- E. Reiter. 1991. Generating Descriptions that Exploit a User's Domain Knowledge. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*, pages 257–285. Academic Press.
- Verena Rieser and Oliver Lemon. 2009. Natural language generation as planning under uncertainty for spoken dialogue systems. In *EACL*.
- Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proceedings of ACL*. (to appear).
- Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Association for Computational Linguistics*.
- Kees van Deemter. 2009. What game theory can do for NLG: the case of vague language. In *12th European Workshop on Natural Language Generation (ENLG)*.
- Marilyn A. Walker, Candace A. Kamm, and Diane J. Litman. 2000. Towards Developing General Models of Usability with PARADISE. *Natural Language Engineering*, 6(3).
- SJ Young, J Schatzmann, K Weilhammer, and H Ye. 2007. The Hidden Information State Approach to Dialog Management. In *ICASSP 2007*.
- S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu. 2009. The Hidden Information State model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*. To appear.



# Helping Our Own: Text Massaging for Computational Linguistics as a New Shared Task

**Robert Dale**

Centre for Language Technology  
Macquarie University  
Sydney, Australia  
Robert.Dale@mq.edu.au

**Adam Kilgarriff**

Lexical Computing Ltd  
Brighton  
United Kingdom  
adam@lexmasterclass.com

## Abstract

In this paper, we propose a new shared task called HOO: Helping Our Own. The aim is to use tools and techniques developed in computational linguistics to help people writing about computational linguistics. We describe a text-to-text generation scenario that poses challenging research questions, and delivers practical outcomes that are useful in the first case to our own community and potentially much more widely. Two specific factors make us optimistic that this task will generate useful outcomes: one is the availability of the ACL Anthology, a large corpus of the target text type; the other is that CL researchers who are non-native speakers of English will be motivated to use prototype systems, providing informed and precise feedback in large quantity. We lay out our plans in detail and invite comment and critique with the aim of improving the nature of the planned exercise.

## 1 Introduction

A forbidding challenge for many scientists whose first language is not English is the writing of acceptable English prose. There is a concern—perhaps sometimes imagined, but real enough to be a worry—that papers submitted to conferences and journals may be rejected because the use of language is jarring and makes it harder for the reader to follow what the author intended. While this can be a problem for native speakers as well, non-native speakers typically face a greater obstacle.

The Association for Computational Linguistics’

mentoring service is one part of a response.<sup>1</sup> A mentoring service can address a wider range of problems than those related purely to writing; but a key motivation behind such services is that an author’s material should be judged on its research content, not on the author’s skills in English.

This problem will surface in any discipline where authors are required to provide material in a language other than their mother tongue. However, as a discipline, computational linguistics holds a privileged position: as scientists, language (of different varieties) is our object of study, and as technologists, language tasks form our agenda. Many of the research problems we focus on could assist with writing problems. There is already existing work that addresses specific problems in this area (see, for example, (Tetreault and Chodorow, 2008)), but to be genuinely useful, we require a solution to the writing problem as a whole, integrating existing solutions to sub-problems with new solutions for problems as yet unexplored.

Our proposal, then, is to initiate a shared task that attempts to tackle the problem head-on; we want to ‘help our own’ by developing tools which can help non-native speakers of English (NNSs) (and maybe some native ones) write academic English prose of the kind that helps a paper get accepted.

The kinds of assistance we are concerned with here go beyond that which is provided by commonly-available spelling checkers and grammar checkers such as those found in Microsoft Word (Heidorn, 2000). The task can be simply expressed as a text-to-text generation exercise:

<sup>1</sup>See <http://acl2010.org/mentoring.htm>.

Given a text, make edits to the text to improve the quality of the English it contains.

This simple characterisation masks a number of questions that must be answered in order to fully specify a task. We turn to these questions in Section 3, after first elaborating on why we think this task is likely to deliver useful results.

## 2 Why This Will Work

### 2.1 Potential Users

We believe this initiative has a strong chance of succeeding simply because there will be an abundance of committed, serious and well-informed users to give feedback on proposed solutions. A familiar problem for technological developments in academic research is that of capturing the time and interest of potential users of the technology, to obtain feedback about what works in a real world task setting, with an appropriate level of engagement.

It is very important to NNS researchers that their papers are not rejected because the English is not good or clear enough. They expect to invest large amounts of time in honing the linguistic aspects of their papers. One of us vividly recalls an explanation by a researcher that, prior to submitting a paper, he took his draft and submitted each sentence in turn, in quotation marks (to force exact matches only), to Google. If there were no Google hits, it was unlikely that the sentence was satisfactory English and it needed reworking; if there were hits, the hits needed checking to ascertain whether they appeared to be written by another non-native speaker.<sup>2</sup> To give that researcher a tool that improves on this situation should not be too great a challenge.

For HOO, we envisage that the researchers themselves, as well as their colleagues, will want to use the prototype systems when preparing their conference and journal submissions. They will have the skills and motivation to integrate the use of prototypes into their paper-writing.

<sup>2</sup>See the Microsoft ESL Assistant at <http://www.eslassistant.com> as an embodiment of a similar idea.

### 2.2 The ACL Anthology

Over a number of years, the ACL has sponsored the ongoing development of the ACL Anthology, a large collection of papers in the domain of computational linguistics. This provides an excellent source for the construction of language models for the task described here. The more recently-prepared ACL Anthology Reference Corpus (Bird et al., 2008), in which 10,921 of the Anthology texts (around 40 million words) have been made available in plain text form, has also been made accessible via the Sketch Engine, a leading corpus query tool.<sup>3</sup>

The corpus is not perfect, of course: not everything in the ACL Anthology is written in flawless English; the ARC was prepared in 2007, so new topics, vocabulary and ideas in CL will not be represented; and the fact that the texts have been automatically extracted from PDF files means that there are errors from the conversion process.

## 3 The Task in More Detail

### 3.1 How Do We Measure Quality?

To be able to evaluate the performance of systems which attempt to improve the quality of a text, we require some means of measuring text quality. One approach would be to develop measures, or make use of existing measures, of characteristics of text quality such as well-formedness and readability (see, for example, (Dale and Chall, 1948; Flesch, 1948; McLaughlin, 1969; Coleman and Liao, 1975)). Given a text and a version of that text that had been subjected to rewriting, we could then compare both texts using these metrics. However, there is always a concern that the metrics may not really measure what they are intended to measure (see, for example, (Le Vie Jr, 2000)); readability metrics have often been criticised for not being good measures of actual readability. The measures also tend to be aggregate measures (for example, providing an average readability level across an entire text), when the kinds of changes that we are interested in evaluating are often very local in nature.

Given these concerns, we opt for a different route: for the initial pilot run of the proposed task, we intend to provide a set of development data consisting

<sup>3</sup>See <http://sketchengine.co.uk/open>.

of 10 conference papers in two versions: an original version of the paper, and an improved version where errors in expression and language use have been corrected. We envisage that participants will focus on developing techniques that attempt to replicate the kinds of corrections found in the improved versions of the papers. For evaluation, we will provide a further ten papers in their original versions, and each participant's results will then be compared against a held-back set of corrected versions for these papers. We would expect the evaluation to assess the following:

- Has the existence of each error annotated in the manually revised versions been correctly identified?
- Have the spans or extents of the errors been accurately identified?
- Has the type of error, as marked in the annotations, been correctly identified?
- How close is the automatically-produced correction to the manually-produced correction?
- What corrections are proposed that do not correspond to errors identified in the manually-corrected text?

With respect to this last point: we anticipate looking closely at all such machine-proposed-errors, since some may indeed be legitimate. Either the human annotators may have missed them, or may not have considered them significant enough to be marked. If there are many such cases, we will need to review how we handle 'prima facie false positives' in the evaluation metrics.

Evaluation of the aspects described above can be achieved automatically; there is also scope, of course, for human evaluation of the overall relative quality of the system-generated texts, although this is of course labour intensive.

### 3.2 Where Does the Source Data Come From?

We have two candidates which we aim to explore as sources of data for the exercise. It is almost certain the first of these two options will yield material which is denser in errors, and closer to the kinds of source material that any practical application will

have to work with; however, the pragmatics of the situation mean that we may have to fall back on our second option.

First, we intend to approach the Mentoring Chairs for the ACL conferences over the last few years with our proposal; then, with their permission, we approach the authors of papers that were submitted for mentoring. If these authors are willing, we use their initial submissions to the mentoring process as the original document set.

If this approach yields an insufficient number of papers (it may be that some authors are not willing to have their drafts made available in this way, and it would not be possible to make them anonymous) then we will source candidate papers from the ACL Anthology. The process we have in mind is this:

- Identify a paper whose authors are non-native English speakers.
- If a quick reading of the paper reveals a moderately high density of correctable errors with in the first page, that paper becomes a candidate for the data set; if it contains very few correctable errors, the paper is ruled as inappropriate.
- Repeat this process until we have a sufficiently large data set.

We then contact the authors to determine whether they are happy for their papers to be used in this exercise. If they are not, the paper is dropped and the next paper's author is asked.

### 3.3 Where do the Corrections Come From?

For the initial pilot, two copy-editors (who may or may not be the authors of this paper) hand-correct the papers in both the development and evaluation data sets. For a full-size exercise there should be more than two such annotators, just as there should be more than ten papers in each of the development and evaluation sets, but our priority here is to test the model before investing further in it.

The copy-editors will then compare corrections, and discuss differences. The possible cases are:

1. One annotator identifies a correction that the other does not.

2. Both annotators identify different corrections for the same input text fragment.

We propose to deal with instances of the first type as follows:

- The two annotators will confer to determine whether one has simply made a mistake—as many authors can testify, no proofreader will find *all* the errors in a text.
- If agreement on the presence or absence of an error cannot be reached, the instance will be dealt with as described below for cases of the second type, with absence of an error being considered a ‘null correction’.

Instances of the second type will be handled as follows:

- If both annotators agree that both alternatives are acceptable, then both alternatives will be provided in the gold standard.
- If no agreement can be reached, then neither alternative will be provided in the gold standard (which effectively means that a null correction is recorded).

Other strategies, such as using a third annotator as a tie-breaker, can be utilised if the task generates a critical mass of interest and volunteer labour.

### 3.4 What Kinds of Corrections?

Papers can go through very significant changes and revisions during the course of their production: large portions of the material can be added or removed, the macro-structure can be re-organised substantially, arguments can be refined or recast. Ideally, a writing advisor might help with large-scale concerns such as these; however, we aim to start at a much simpler level, focussing on what is sometimes referred to as a ‘light copy-edit’. This involves a range of phenomena which can be considered sentence-internal:

- domain- and genre-specific spelling errors, including casing errors;
- dispreferred or suboptimal lexical choices;

- basic grammatical errors, including common ESL problems like incorrect preposition and determiner usage;
- reduction of syntactic complexity;
- stylistic infelicities which, while not grammatically incorrect, are unwieldy and impact on fluency and ease of reading.

The above are all identifiable and correctable within the context of a single sentence; however, we also intend to correct inconsistencies across the document as whole:

- consistency of appropriate tense usage;
- spelling and hyphenation instances where there is no obvious correct answer, but a uniformity is required.

We envisage that the process of marking up the gold-standard texts will allow us to develop more formal guidelines and taxonomic descriptions for use subsequent to the pilot exercise. There are, of course, existing approaches to error markup that can provide a starting point here, in particular the schemes used in the large-scale exercises in learner error annotation undertaken at CECL, Louvain-la-Neuve (Dagneaux et al., 1996) and at Cambridge ESOL (Nicholls, 2003).

### 3.5 How Should the Task be Approached?

There are many ways in which the task could be addressed; it is open to both rule-based and statistical solutions. An obvious way to view the task is as a machine translation problem from poor English to better English; however, supervised machine learning approaches may be ruled out by the absence of an appropriately large training corpus, something we may not see until the task has generated significant momentum (or more volunteer annotators at an early stage!).

There is clearly a wealth of existing research on grammar and style checking that can be brought to bear. Although grammar and style checking has been in the commercial domain now for three decades, the task may provide a framework for the first comparative test of many of these applications.



Because the nature of errors is so diverse, this task offers the opportunity to exercise a broad range of approaches to the problem, and also allows for narrowly-focussed solutions that attempt to address specific problems with high accuracy.

#### 4 Some Potential Problems

Our proposal is not without possible problems and detrimental side effects.

Clearly there are ethical issues that need to be considered carefully; even if an author is happy for their data to be used in this way, one might find retrospective embarrassment at eponymous error descriptions entering the common vocabulary in the field—it's one thing to be acknowledged for Kneser-Ney smoothing, but perhaps less appealing to be famous for the Dale-Kilgariff adjunct error.

Our suggestion that the ACL Anthology might be used as a source for language modelling brings its own downsides: in particular, if anything is likely to increase the oft-complained-about sameness of CL papers, this will! There is also an ethical issue around the fine line between what such systems will do and plagiarism; one might foresee the advent of a new scholastic crime labelled 'machine-assisted style plagiarism'.

There are no doubt other issues we have not yet considered; again, feedback on potential pitfalls is eagerly sought.

#### 5 Next Steps

Our aim is to obtain feedback on this proposal from conference participants and others, with the aim of refining our plan in the coming months. If we sense that there is a reasonable degree of interest in the task, we would aim to publish the initial data set well before the end of the year, with a first evaluation taking place in 2011.

In the name of better writing, CLers of the world unite—you have nothing to lose but your worst sentences!

#### Acknowledgements

We thank the two anonymous reviewers for useful feedback on this proposal, and Anja Belz for encouraging us to develop the idea.

#### References

- Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2008)*, location = Marrakesh, Morocco.
- Meri Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60:283–284.
- E Dagneaux, S Denness, S Granger, and F Meunier. 1996. Error tagging manual version 1.1. Technical report, Centre for English Corpus Linguistics, Université Catholique de Louvain.
- Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational research bulletin*, 27:11–20.
- Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233.
- George Heidorn. 2000. Intelligent writing assistance. In R Dale, H Moisl, and H Somers, editors, *Handbook of Natural Language Processing*, pages 181–207. Marcel Dekker Inc.
- Donald S. Le Vie Jr. 2000. Documentation metrics: What do you really want to measure? *Intercom*.
- G. Harry McLaughlin. 1969. SMOG grading – a new readability formula. *Journal of Reading*, pages 639–646.
- D Nicholls. 2003. The cambridge learner corpus: error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 Conference (CL 2003)*, page 572.
- J R Tetreault and M S Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics*.



# Finding Common Ground: Towards a Surface Realisation Shared Task

**Anja Belz**

Natural Language Technology Group  
Computing, Mathematical and Information Sciences  
University of Brighton, Brighton BN2 4GJ, UK  
a.s.belz@brighton.ac.uk

**Mike White**

Department of Linguistics  
The Ohio State University  
Columbus, OH, USA  
mwhite@ling.osu.edu

**Josef van Genabith and Deirdre Hogan**

National Centre for Language Technology  
School of Computing  
Dublin City University  
Dublin 9, Ireland  
{dhogan, josef}@computing.dcu.ie

**Amanda Stent**

AT&T Labs Research, Inc.,  
180 Park Avenue  
Florham Park, NJ 07932, USA  
stent@research.att.com

## Abstract

In many areas of NLP reuse of utility tools such as parsers and POS taggers is now common, but this is still rare in NLG. The subfield of surface realisation has perhaps come closest, but at present we still lack a basis on which different surface realisers could be compared, chiefly because of the wide variety of different input representations used by different realisers. This paper outlines an idea for a shared task in surface realisation, where inputs are provided in a common-ground representation formalism which participants map to the types of input required by their system. These inputs are derived from existing annotated corpora developed for language analysis (parsing etc.). Outputs (realisations) are evaluated by automatic comparison against the human-authored text in the corpora as well as by human assessors.

## 1 Background

When reading a paper reporting a new NLP system, it is common these days to find that the authors have taken an NLP utility tool off the shelf and reused it. Researchers frequently reuse parsers, POS-taggers, named entity recognisers, coreference resolvers, and many other tools. Not only is there a real choice between a range of different systems performing the same task, there are also evaluation methodologies to help determine what the state of the art is.

Natural Language Generation (NLG) has not

so far developed generic tools and methods for comparing them to the same extent as Natural Language Analysis (NLA) has. The subfield of NLG that has perhaps come closest to developing generic tools is surface realisation. Wide-coverage surface realisers such as PENMAN/NIGEL (Mann and Mathiesen, 1983), FUF/SURGE (Elhadad and Robin, 1996) and REALPRO (Lavoie and Rambow, 1997) were intended to be more or less off-the-shelf plug-and-play modules. But they tended to require a significant amount of work to adapt and integrate, and required highly specific inputs incorporating up to several hundred features that needed to be set.

With the advent of statistical techniques in NLG surface realisers appeared for which it was far simpler to supply inputs, as information not provided in the inputs could be added on the basis of likelihood. An early example, the Japan-Gloss system (Knight et al., 1995) replaced PENMAN's default settings with statistical decisions. The Halogen/Nitrogen developers (Langkilde and Knight, 1998a) allowed inputs to be arbitrarily underspecified, and any decision not made before the realiser was decided simply by highest likelihood according to a language model, automatically trainable from raw corpora.

The Halogen/Nitrogen work sparked an interest in statistical NLG which led to a range of surface realisation methods that used corpus frequencies in one way or another (Varges and Mellish, 2001; White, 2004; Vellidal et al., 2004; Paiva and Evans, 2005). Some surface realisation work looked at directly applying statistical models during a linguistically informed generation process to prune

the search space (White, 2004; Carroll and Oepen, 2005).

While statistical techniques have led to realisers that are more (re)usable, we currently still have no way of determining what the state of the art is. A significant subset of statistical realisation work (Langkilde, 2002; Callaway, 2003; Nakanishi et al., 2005; Zhong and Stent, 2005; Cahill and van Genabith, 2006; White and Rajkumar, 2009) has recently produced results for regenerating the Penn Treebank. The basic approach in all this work is to remove information from the Penn Treebank parses (the word strings themselves as well as some of the parse information), and then convert and use these underspecified representations as inputs to the surface realiser whose task it is to reproduce the original treebank sentence. Results are typically evaluated using BLEU, and, roughly speaking, BLEU scores go down as more information is removed.

While publications of work along these lines do refer to each other and (tentatively) compare BLEU scores, the results are not in fact directly comparable, because of the differences in the input representations automatically derived from Penn Treebank annotations. In particular, the extent to which they are underspecified varies from one system to the next.

The idea we would like to put forward with this short paper is to develop a shared task in surface realisation based on common inputs and annotated corpora of paired inputs and outputs derived from various resources from NLA that build on the Penn Treebank. Inputs are provided in a common-ground representation formalism which participants map to the types of input required by their system. These inputs are automatically derived from the Penn Treebank and the various layers of annotation (syntactic, semantic, discourse) that have been developed for the documents in it. Outputs (realisations) are evaluated by automatic comparison against the human-authored text in the corpora as well as by human assessors.

In the short term, such a shared task would make existing and new approaches directly comparable by evaluation on the benchmark data associated with the shared task. In the long term, the common-ground input representation may lead to a standardised level of representation that can act as a link between surface realisers and preceding modules, and can make it possible to use alterna-

tive surface realisers as drop-in replacements for each other.

## 2 Towards Common Inputs

One hugely challenging aspect in developing a Surface Realisation task is developing a common input representation that all, or at least a majority of, surface realisation researchers are happy to work with. While many different formalisms have been used for input representations to surface realisers, one cannot simply use e.g. van Genabith et al.'s automatically generated LFG f-structures, White et al.'s CCG logical forms, Nivre's dependencies, Miyao et al.'s HPSG predicate-argument structures or Copestake's MRSS etc., as each of them would introduce a bias in favour of one type of system.

One possible solution is to develop a meta-representation which contains, perhaps on multiple layers of representation, all the information needed to map to any of a given set of realiser input representations, a common-ground representation that acts as a kind of interlingua for translating between different input representations.

An important issue in deriving input representations from semantically, syntactically and discourse-annotated corpora is deciding what information *not* to include. A concern is that making such decisions by committee may be difficult. One way to make it easier might be to define several versions of the task, where each version uses inputs of different levels of specificity.

Basing a common input representation on what can feasibly be obtained from non-NLG resources would put everyone on reasonably common footing. If, moreover, the common input representations can be automatically derived from annotations in existing resources, then data can be produced in sufficient quantities to make it feasible for participants to automatically learn mappings from the system-neutral input to their own input.

The above could be achieved by doing something along the lines of the CONLL'08 shared task on Joint Parsing of Syntactic and Semantic Dependencies, for which the organisers combined the Penn Treebank, Propbank, Nombank and the BBN Named Entity corpus into a dependency representation. Brief descriptions of these resources and more details on this idea are provided in Section 4 below.

### 3 Evaluation

As many NLG researchers have argued, there is usually not a single right answer in NLG, but various answers, some better than others, and NLG tasks should take this into account. If a surface realisation task is focused on single-best realizations, then it will not encourage research on producing all possible good realizations, or multiple acceptable realizations in a ranked list, etc. It may not be the best approach to encourage systems that try to make a single, safe choice; instead, perhaps one should encourage approaches that can tell when multiple choices would be ok, and if some would be better than others.

In the long term we need to develop task definitions, data resources and evaluation methodologies that properly take into account the one-to-many nature of NLG, but in the short term it may be more realistic to reuse existing non-NLG resources (which do not provide alternative realisations) and to adapt existing evaluation methodologies including intrinsic assessment of Fluency, Clarity and Appropriateness by trained evaluators, and automatic intrinsic methods such as BLEU and NIST. One simple way of adapting the latter, for example, could be to calculate scores for the  $n$  best realisations produced by a realiser and then to compute a weighted average where scores for realisations are weighted in inverse proportion to the ranks given to the realisations by the realiser.

### 4 Data

There is a wide variety of different annotated resources that could be of use in a shared task in surface realisation. Many of these include documents originally included in the Penn Treebank, and thus make it possible in principle to combine the various levels of annotation into a single common-ground representation. The following is a (non-exhaustive) list of such resources:

1. Penn Treebank-3 (Marcus et al., 1999): one million words of hand-parsed 1989 Wall Street Journal material annotated in Treebank II style. The Treebank bracketing style allows extraction of simple predicate/argument structure. In addition to Treebank-1 material, Treebank-3 contains documents from the Switchboard and Brown corpora.
2. Propbank (Palmer et al., 2005): This is a semantic annotation of the Wall Street Journal

section of Penn Treebank-2. More specifically, each verb occurring in the Treebank has been treated as a semantic predicate and the surrounding text has been annotated for arguments and adjuncts of the predicate. The verbs have also been tagged with coarse grained senses and with inflectional information.

3. NomBank 1.0 (Meyers et al., 2004): NomBank is an annotation project at New York University that provides argument structure for common nouns in the Penn Treebank. NomBank marks the sets of arguments that occur with nouns in PropBank I, just as the latter records such information for verbs.
4. BBN Pronoun Coreference and Entity Type Corpus (Weischedel and Brunstein, 2005): supplements the Wall Street Journal corpus, adding annotation of pronoun coreference, and a variety of entity and numeric types.
5. FrameNet (Johnson et al., 2002): 150,000 sentences annotated for semantic roles and possible syntactic realisations. The annotated sentences come from a variety of sources, including some PropBank texts.
6. OntoNotes 2.0 (Weischedel et al., 2008): OntoNotes 1.0 contains 674k words of Chinese and 500k words of English newswire and broadcast news data. OntoNotes follows the Penn Treebank for syntax and PropBank for predicate-argument structure. Its semantic representation will include word sense disambiguation for nouns and verbs, with each word sense connected to an ontology, and coreference. The current goal is to annotate over a million words each of English and Chinese, and half a million words of Arabic over five years.

There are other resources which may be useful. Zettlemoyer and Collins (2009) have manually converted the original SQL meaning annotations of the ATIS corpus (et al., 1994)—some 4,637 sentences—into lambda-calculus expressions which were used for training and testing their semantic parser. This resource might make a good out-of-domain test set for generation systems trained on WSJ data.

FrameNet, used for semantic parsing, see for example Gildea and Jurafsky (2002), identifies a

sentence's frame elements and assigns semantic roles to the frame elements. FrameNet data (Baker and Sato, 2003) was used for training and test sets in one of the SensEval-3 shared tasks in 2004 (Automatic Labeling of Semantic Roles). There has been some work combining FrameNet with other lexical resources. For example, Shi and Mihalcea (2005) integrated FrameNet with VerbNet and WordNet for the purpose of enabling more robust semantic parsing.

The Semlink project (<http://verbs.colorado.edu/semLink/>) aims to integrate Propbank, FrameNet, WordNet and VerbNet.

Other relevant work includes Moldovan and Rus (Moldovan and Rus, 2001; Rus, 2002) who developed a technique for parsing into logical forms and used this to transform WordNet concept definitions into logical forms. The same method (with additional manual correction) was used to produce the test set for another SensEval-3 shared task (Identification of Logic Forms in English).

#### 4.1 CoNLL 2008 Shared Task Data

Perhaps the most immediately promising resource is the CoNLL shared task data from 2008 (Surdanu et al., 2008) which has syntactic dependency annotations, named-entity boundaries and the semantic dependencies model roles of both verbal and nominal predicates. The data consist of excerpts from Penn Treebank-3, BBN Pronoun Coreference and Entity Type Corpus, PropBank I and NomBank 1.0. In CoNLL '08, the data was used to train and test systems for the task of producing a joint semantic and syntactic dependency analysis of English sentences (the 2009 CoNLL Shared Task extended this to multi-lingual data).

It seems feasible that we could reuse the CoNLL data for a prototype Surface Realisation task, adapting it and inverting the direction of the task, i.e. mapping from syntactic-semantic dependency representations to word strings.

### 5 Developing the Task

The first step in developing a Surface Realisation task could be to get together a working group of surface realisation researchers to develop a common-ground input representation automatically derivable from a set of existing resources. As part of this task a prototype corpus exemplifying inputs/outputs and annotations could be developed. At the end of this stage it would be use-

ful to write a white paper and circulate it and the prototype corpus among the NLG (and wider NLP) community for feedback and input.

After a further stage of development, it may be feasible to run a prototype surface realisation task at Generation Challenges 2011, combined with a session for discussion and roadmapping. Depending on the outcome of all of this, a full-blown task might be feasible by 2012. Some of this work will need funding to be feasible, and the authors of this paper are in the process of applying for financial support for these plans.

### 6 Concluding Remarks

In this paper we have provided an overview of existing resources that could potentially be used for a surface realisation task, and have outlined ideas for how such a task might work. The core idea is to develop a common-ground input representation which participants map to the types of input required by their system. These inputs are derived from existing annotated corpora developed for language analysis. Outputs (realisations) are evaluated by automatic comparison against the human-authored text in the corpora as well as by human assessors. Evaluation methods are adapted to take account of the one-to-many nature of the realisation mapping.

The ideas outlined in this paper began as a prolonged email exchange, interspersed with discussions at conferences, among the authors. This paper summarises our ideas as they have evolved so far, to enable feedback and input from other researchers interested in this type of task.

### References

- Colin F. Baker and Hiroaki Sato. 2003. The framenet data and software. In *Proceedings of ACL'03*.
- A. Cahill and J. van Genabith. 2006. Robust PCFG-based generation using automatically acquired LFG approximations. In *Proc. ACL'06*, pages 1033–44.
- Charles Callaway. 2003. Evaluating coverage for large symbolic NLG grammars. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 811–817.
- J. Carroll and S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)*, volume 3651, pages 165–176. Springer Lecture Notes in Artificial Intelligence.

- M. Elhadad and J. Robin. 1996. An overview of SURGE: A reusable comprehensive syntactic realization component. Technical Report 96-03, Dept of Mathematics and Computer Science, Ben Gurion University, Beer Sheva, Israel.
- Deborah Dahl et al. 1994. Expanding the scope of the ATIS task: the ATIS-3 corpus. In *Proceedings of the ARPA HLT Workshop*.
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.
- C. Johnson, C. Fillmore, M. Petruck, C. Baker, M. Ellsworth, J. Ruppenhoper, and E. Wood. 2002. Framenet theory and practice. Technical report.
- K. Knight, I. Chander, M. Haines, V. Hatzivassiloglou, E. Hovy, M. Iida, S. Luk, R. Whitney, and K. Yamada. 1995. Filling knowledge gaps in a broad-coverage MT system. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 1390–1397.
- I. Langkilde and K. Knight. 1998a. Generation that exploits corpus-based statistical knowledge. In *Proc. COLING-ACL*. <http://www.isi.edu/licensed-sw/halogen/nitro98.ps>.
- I. Langkilde. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. 2nd International Natural Language Generation Conference (INLG '02)*.
- B. Lavoie and O. Rambow. 1997. A fast and portable realizer for text generation systems. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP'97)*, pages 265–268.
- W. Mann and C. Mathiesen. 1983. NIGEL: A systemic grammar for text generation. Technical Report ISI/RR-85-105, Information Sciences Institute.
- Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3. Technical report, Linguistic Data Consortium, Philadelphia.
- Adam Meyers, Ruth Reeves, and Catherine Macleod. 2004. Np-external arguments a study of argument sharing in english. In *MWE '04: Proceedings of the Workshop on Multiword Expressions*, pages 96–103, Morristown, NJ, USA. Association for Computational Linguistics.
- Dan I. Moldovan and Vasile Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of ACL'01*.
- Hiroko Nakanishi, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic models for disambiguation of an hpsg-based chart generator. In *Proceedings of the 9th International Workshop on Parsing Technology (Parsing'05)*, pages 93–102. Association for Computational Linguistics.
- D. S. Paiva and R. Evans. 2005. Empirically-based control of natural language generation. In *Proceedings ACL'05*.
- M. Palmer, P. Kingsbury, and D. Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Vasile Rus. 2002. *Logic Form For WordNet Glosses and Application to Question Answering*. Ph.D. thesis.
- Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Proceedings of CILing'05*.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL '08: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177.
- S. Varges and C. Mellish. 2001. Instance-based natural language generation. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL '01)*, pages 1–8.
- E. Veldal, S. Oepen, and D. Flickinger. 2004. Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT '04)*, Tuebingen, Germany.
- Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. Technical report, Linguistic Data Consortium.
- Ralph Weischedel et al. 2008. Ontonotes release 2.0. Technical report, Linguistic Data Consortium.
- Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for ccg realisation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*, pages 410–419.
- M. White. 2004. Reining in CCG chart realization. In A. Belz, R. Evans, and P. Piwek, editors, *Proceedings INLG'04*, volume 3123 of *LNAI*, pages 182–191. Springer.
- Luke Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical forms. In *Proceedings of ACL-IJCNLP'09*.
- H. Zhong and A. Stent. 2005. Building surface realizers automatically from corpora. In A. Belz and S. Varges, editors, *Proceedings of UCNLG'05*, pages 49–54.

## References

- John A. Bateman. 1999. Using aggregation for selecting content when generating referring expressions. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL'99)*, pages 127–134, Morristown, NJ, USA.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Keith Devlin. 2006. Situation theory and situation semantics. In Dov M. Gabbay and John Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 601–664. Elsevier.
- Paul E. Engelhardt, Karl G.D. Bailey, and Fernanda Ferreira. 2006. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, 54(4):554–573.
- Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. 2004. Generation of relative referring expressions based on perceptual grouping. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA.
- Simon Garrod and Martin J. Pickering. 2004. Why is conversation so easy? *Trends in Cognitive Sciences*, 8(1):8–11, January.
- Stephen C. Hirtle and John Jonides. 1985. Evidence for hierarchies in cognitive maps. *Memory and Cognition*, 13:208–217.
- Helmut Horacek. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-97)*, pages 206–213, Morristown, NJ, USA.
- Emiel Krahmer and Mariët Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and R. Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford, CA, USA.
- Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- Benjamin Kuipers. 1977. *Representing Knowledge of Large-scale Space*. PhD thesis, MIT-AI TR-418, Massachusetts Institute of Technology, Cambridge, MA, USA, May.
- Timothy P. McNamara. 1986. Mental representations of spatial relations. *Cognitive Psychology*, 18:87–121.
- Ivandr  Paraboni, Kees van Deemter, and Judith Masthoff. 2007. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254, June.
- Massimo Poesio. 1993. A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues. In Peter Aczel, David Israel, Yasuhiro Katagiri, and Stanley Peters, editors, *Situation Theory and its Applications Volume 3*, CSLI Lecture Notes No. 37, pages 339–374. Center for the Study of Language and Information, Menlo Park, CA, USA.
- Albert Stevens and Patty Coupe. 1978. Distortions in judged spatial relations. *Cognitive Psychology*, 10:422–437.
- Kees van Deemter. 2002. Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52.
- Jette Viethen and Robert Dale. 2006. Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 63–70, Sydney, Australia.
- Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayova. 2009a. A situated context model for resolution and generation of referring expressions. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 126–129, Athens, Greece, March.
- Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayova. 2009b. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1604–1609, Pasadena, CA, USA, July.



# Appendices



## Author Index

- Bandyopadhyay, Sivaji ..... 235  
Belz, Anja ..... 7, 167, 217, 219, 267  
Bohnet, Bernd ..... 237  
Bouchard, Guillaume ..... 230  
Brennan, Susan E. .... 2  
Byron, Donna ..... 243  
  
Carberry, Sandra ..... 17  
Cassell, Justine ..... 243  
Carenini, Giuseppe ..... 105  
Charton, Éric ..... 233  
Cristià, Maximiliano ..... 173  
Cuayahuitl, Heriberto ..... 37  
  
Dale, Robert ..... 243, 261  
Dannélls, Dana ..... 179  
Das, Amitava ..... 235  
de Kok, Daniël ..... 155  
Demir, Seniz ..... 17  
Denis, Alexandre ..... 27  
Dethlefs, Nina ..... 37  
  
Elson, David ..... 47  
Favre, Benoit ..... 237  
Foster, Mary Ellen ..... 67  
  
Gagnon, Michel ..... 233  
Gargett, Andrew ..... 243  
Gatt, Albert ..... 57, 217  
Giuliani, Manuel ..... 67  
Greenbacker, Charlie ... 185, 239, 241  
Greeve, Fai ..... 209  
  
Hogan, Deirdre ..... 267  
  
Iida, Ryu ..... 135  
Isard, Amy ..... 67  
  
Janarthanam, Srini ..... 255  
  
Kilgarriff, Adam ..... 261  
Knoll, Alois ..... 67  
Koller, Alexander ..... 217, 243  
Koolen, Ruud ..... 191  
Koppermann, Christopher ..... 209  
  
Kow, Eric ..... 7, 167, 219  
Krahmer, Emiel ..... 191, 203  
Kruijff, Geert-Jan ..... 209  
Kuo, Che-Yu ..... 239, 241  
Kuriyama, Naoko ..... 135  
  
Lemon, Oliver ..... 255  
Lintean, Mihai ..... 251  
  
Matheson, Colin ..... 67  
McDonald, David ..... 185  
McKeown, Kathleen ..... 47  
McCoy, Kathleen F. .... 17, 239, 241  
Mellish, Chris ..... 77, 85  
Mitchell, Margaret ..... 95  
Moldovan, Christian ..... 251  
Mondal, Tapabrata ..... 235  
Moore, Johann ..... 243  
Murray, Gabriel ..... 105  
  
Ng, Raymond ..... 105  
  
Oberlander, Jon ..... 67, 243  
Ozell, Benoit ..... 233  
  
Pan, Jeff Z. .... 115  
Piwek, Paul ..... 145, 251  
Plüss, Brian ..... 173  
Portet, Francois ..... 57  
Power, Richard ..... 3, 197  
  
Rieser, Verena ..... 255  
Reiter, Ehud ..... 95  
Ren, Yuan ..... 115  
Rus, Vasile ..... 251  
  
Saikh, Tanik ..... 235  
Siddharthan, Advaith ..... 125  
Spanger, Philipp ..... 135  
Sparks, Nicole ..... 239, 241  
Stent, Amanda ..... 267  
Stoyanchev, Svetlana ..... 145, 251  
Striegnitz, Kristina ..... 243

Terai, Asuka .....	135
Theune, Mariet .....	191
Tokunaga, Takenobu .....	135
van Deemter, Kees .....	95, 115
van den Bosch, Antal .....	203
van Genabith, Josef .....	267
White, Mike .....	267
Williams, Sandra .....	197
Wubben, Sander .....	203
Wyse, Brendan .....	251
Zender, Hendrik .....	209





Hosted by the University of Dublin, Trinity College and the Dublin Institute of Technology  
Endorsed by the ACL Special Interest Group on Natural Language Generation (SIGGEN)  
Sponsored by the Centre for Next Generation Localisation and Science Foundation Ireland