DAVID M. HOWCROFT

LEARNING TO GENERATE

LEARNING TO GENERATE

DAVID M. HOWCROFT

Bayesian nonparametric approaches to inducing rules for natural language generation

August 2021

Ubi caritas et amor
Deus ibi ist

Where there is charity and love,
there God is.

ABSTRACT

In order for computers to produce natural language texts from non-linguistic information, we need a system for mapping between the two, a system of Natural Language Generation (NLG). We can reduce the difficulty of developing such systems if we leverage Machine Learning (ML) intelligently. While there are many possible approaches to the task, this thesis argues for one in particular, focusing on sentence planning using synchronous grammars and Bayesian nonparametric methods.

We formulate sentence planning rules in terms of Synchronous Tree Substitution Grammars (sTSGs) and implement a series of hierarchical Dirichlet Processes along with a Gibbs sampler to learn such rules from appropriate corpora. Due to the lack of corpora which pair hierarchical, discourse-structured meaning representations with varied texts, we developed a new interface for crowdsourcing training corpora for NLG systems by asking participants to produce paraphrases of pre-existing texts and collected a new corpus, which we call the Extended SPaRKy Restaurant Corpus (ESRC).

After training our models on pre-existing, lexically-restricted corpora as well as the ESRC, we conduct a series of human evaluations using a novel evaluation interface. This interface enables the assessment of the fluency, semantic fidelity, and expression of discourse relations in a text in a single crowdsourcing experiment. While we identify several limitations to our approach, the evaluations suggest that our models can outperform existing neural network models with respect to semantic fidelity and in some cases maintain similar levels of fluency.

In addition to these efforts, we present a Dependency Attachment Grammar (DAG) based on (Joshi & Rambow, 2003) and extend this grammar to the synchronous setting so that future work can build upon its added flexibility relative to sTSG. In addition to these practically-oriented efforts, we also explore human variation in adapting their utterances to listeners under cognitive load through a psycholinguistic study.

This thesis opens up several directions for future research into how best to integrate the various challenging tasks involved in natural language generation and how best to evaluate these systems in the future.

ZUSAMMENFASSUNG

Damit Computer aus nicht-linguistischen Informationen natürlichsprachliche Texte erzeugen können, brauchen wir ein System zur Zuordnung zwischen den beiden, ein sogenanntes „Natural Language Generation" (NLG) System. Wir können die Entwicklung solcher Systeme vereinfachen, wenn wir Maschinelles Lernen (ML) intelligent nutzen. Obwohl es viele mögliche Herangehensweisen an die Aufgabe gibt, spricht sich diese Dissertation insbesondere für einen aus, der sich auf die Satzplanung mit synchronen Grammatiken und Bayesschen nichtparametrischen Methoden konzentriert.

Wir haben eine auf synchroner Baumersetzungsgrammatik (sTSG)[1] basierende Darstellung für Satzpläne entwickelt und ein Modell zum Erlernen dieser Regeln mit hierarchischen Dirichlet-Prozessen implementiert. Um unser Modell zu trainieren, haben wir ein neuartiges Korpus erstellt, das gepaarte Texte und Diskursstrukturen enthält, und um Instanziierungen des Modells zu evaluieren, haben wir eine neuartige Umfrage-Schnittstelle für menschliche Bewertungen entwickelt. Wir schließen auch eine psycholinguistische Studie ein, die hilft, unser Interesse an sprachlicher Variation zu begründen.

DARSTELLUNGEN FÜR SATZPLANUNGSREGELN

Satzplanungsregeln werden verwendet, um von einer (pseudo-) semantischen Eingabedarstellung (ein Textplan, TP) auf eine (pseudo-) syntaktische Ausgabedarstellung (eine logische Form, LF) abzubilden, wobei Lexikalisierung, Aggregation und Generation von Referenzausdrücken durchgeführt werden. In Kapitel 3 beschrieben wir, wie synchrone Baumersetzungsgrammatik (sTSG) verwendet werden könnte, um solche Regeln auszudrücken, und haben die formalen Darstellungen erstellt, die erforderlich sind, um Modelle basierend auf solchen Grammatiken zu implementieren. Wir bauten auf diesen Darstellungen auf, um eine formale Definition einer Grammatik für Dependenzbäume mit Anhang ähnlich wie (Joshi & Rambow, 2003) bereitzustellen, die wir „Dependency Attachment Grammar" nennen,[2] und erweiterten diese Grammatik auf die synchrone Umgebung.

Während Baumersetzungs- und Baumadjunktions-Grammatiken (TSG und TAG) seit Jahrzehnten im Kontext von NLG diskutiert werden, ist die aktuelle Arbeit die erste, die speziell vorschlägt, synchronen Baumersetzungsgrammatik (sTSG) zur Darstellung von Satzplanungsregeln zu verwenden. Dieser Rahmen erleichtert die Einbeziehung

1 Aus dem Englischen „synchronous Tree Substitution Grammar"
2 „Dependenzverbindungsgrammatik" auf Deutsch

semantischer Domänen und allgemeiner linguistischer Kenntnisse in Lernmodelle. Darüber hinaus sind wir die ersten, die synchrone „Dependency Attachment Grammar" beschreiben, die unserer Meinung nach verwendet werden kann, um in zukünftigen Arbeiten über die Möglichkeiten von sTSG zur Darstellung von Satzplanungsregeln hinauszugehen.

MODELLE ZUM ERLERNEN VON SATZPLANUNGSREGELN

Im Kapitel 6 wurde insbesondere die Motivation zum Erlernen von Satzplanungsregeln erläutert. Durch die Zerlegung der NLG-Aufgabe nach dem Vorbild der traditionellen NLG-Architektur können wir die Lernaufgabe vereinfachen, da das Modell nicht mehr jede Aufgabe auf einmal lernen muss. Darüber hinaus ermöglicht es uns die Konzentration auf das Erlernen von Satzplanungsregeln, vorhandene Systeme für die Oberflächenrealisierung zu nutzen, sodass unser System die morphologischen, Linearisierungs-, Großschreibungs- und Interpunktionseigenheiten der Sprache, die es generiert, nicht lernen muss.

In diesem Kapitel wurden auch weitere Implementierungsdetails in Bezug auf die Regelanwendung und Oberflächenrealisierung erläutert, sodass sich Kapitel 9 stattdessen auf das von uns implementierte spezielle ML-Modell konzentrieren konnte, aufbauend auf früheren Arbeiten zur Grammatik Induktion für sTSG (siehe Kapitel 5). In Kapitel 9 entwickelten wir einen hierarchischen Dirichlet-Prozess, um TSG für Dependenzbäume zu modellieren, bevor wir diese Modelle unter einem anderen Dirichlet-Prozess miteinander verbinden, um ein sTSG für Satzplanungsregeln zu modellieren. Wir haben eine Reihe sogenannter Gibbs-Operatoren eingeführt, um Modellaktualisierungen basierend auf unseren Trainingsdaten durchzuführen und die Mischung gegenüber einfacheren Segmentierungsmodellen zu verbessern. Wir untersuchen zwei Ansätze zur Initialisierung der Alignments zwischen TP und LF und stellen fest, dass der einfachere Ansatz in unseren Evaluierungen zu einer besseren Systemleistung führte.

Um unser Modell zu testen, führten wir drei Versuchsreihen durch. In der ersten haben wir das Modell auf eine Testset angewendet, das aus einem Korpus von Eingabe-Ausgabe-Paaren eines bestehenden NLG-Systems (dem „SPaRKy Restaurant Corpus", SRC Walker u. a., 2007) stammt. Dies hilft zu identifizieren, welche Modellparameter für die Fähigkeit des Systems relevant sind, Outputs für eine gegebene Menge von Inputs zu erzeugen (dh sein Umfang) und welche Modellparameter zu wesentlichen Änderungen in den Outputtexten führten (dh Outputähnlichkeit).

Bei der Beurteilung der Qualität dieser Texte durch Menschen stellten wir fest, dass die sogenannte „Fluency"[3] unseres Modells der einer modernen Basislinie eines neuronalen Netzwerks ähnlich ist, während es in Bezug auf die semantische Genauigkeit wesentlich besser abschneidet, weniger Fakten in der Eingabe ausgelassen und die beabsichtigte Reihenfolge des Ausdrucks beibehalten wurde.

Das Testset, das für die erste menschliche Auswertung verwendet wurde, bestand jedoch fast ausschließlich aus JUSTIFICATION-Relationen, daher haben wir auch einen neuen Satz von TP generiert, der nur CONTRAST-Relationen enthält, und führten eine zweite Studie durch. Diese Studie hat eine Schwäche unseres Ansatzes aufgezeigt, da die Umfang für unser Modell bei dieser neuen Menge von TP erheblich gesunken ist. Für diejenigen TP, bei denen wir einen Text generieren konnten, ergab unsere menschliche Beurteilung jedoch eine mit unserer Baseline vergleichbare „Fluency", während Auslassungen vermieden und die inhaltliche Reihenfolge beibehalten wurde.

Während dieses Experiment neuartige Testdaten verwendete, wurden keine naturalistischen Korpusdaten verwendet, wie sie ein Forscher mit menschlichen Teilnehmern sammeln könnte. Daher konzentrierte sich unser drittes Experiment auf unseren Datensatz, den „Extended SPaRKy Restaurant Corpus" (ESRC), der Texte mit höherer und niedrigerer Informationsdichte enthält.

Weder unsere Baseline noch unser Modell schnitten bei diesem Datensatz besonders gut ab, wobei die „Fluency" und die semantische Genauigkeit bei beiden dramatisch abfielen, obwohl die Inhaltsreihenfolge bei unseres System immer noch besser als beim Baseline-System erhalten blieb. Mit diesem Datensatz konnten wir jedoch vollumfänglich beobachten, wie sich das Vertrauen auf ein bestehendes, regelbasiertes System mit einer auf einer bestimmten Domäne basierenden Grammatik auswirkt: unser Oberflächenrealisierer verwendete eine Grammatik aus der Zeitungsdomäne und nicht die informelle schriftliche Domäne aus dem ESRC. Aufgrund der schlechten Textqualität bei der Auswertung mit dem ESRC, sind wir nicht in der Lage, die Fähigkeit des Modells, die im zugrunde liegenden Korpus vorhandene Variation zu emulieren, weiter zu untersuchen.

Das in dieser Arbeit vorgestellte System ist das erste System zur Grammatikinduktion für die Satzplanung im Besonderen und für synchrone Dependenzbäume im Allgemeinen. Unsere Bewertungen zeigen Schwächen auf, die auf der Gesamtpipeline basieren, in der sich das Modell befindet, und zeigen gleichzeitig, dass der Ansatz im Allgemeinen gute Arbeit leistet, um semantische Inhalte zu erhalten und richtig zu ordnen. Dies legt nahe, dass es sich lohnt, in zukünftigen Arbeiten alternative Implementierungen zu untersuchen (siehe Abschnitt 11.5.1).

3 Hiermit meinen wir etwas wie ‚sprachflüssig' oder ‚sprachgewandt'. Kapitel 8 diskutiert Probleme mit den Definitionen solcher Aspekte eines Textes.

NEUE DATENSÄTZE UND SPRACHLICHE VARIATION

In Kapitel 7 haben wir aktuelle Korpora für NLG und die Desiderate zum Trainieren unserer Satzplanungsmodelle untersucht. Frühere Korpora enthielten begrenzte Diskursinformationen, die Eingaben nicht als Textpläne, sondern als eine Sammlung von Schlüssel-Wert-Paaren darstellten, die den auszudrückenden Fakten entsprechen, oder enthielten begrenzte Variationen, die auf den Ausgaben bestehender NLG-Systeme auf einem begrenzten Regelwerk basieren.

Daher haben wir ein neuartiges Paraphrasierungsparadigma für die Crowdsourcing-Datensammlung entwickelt, um unsere Modelle an einem Korpus zu trainieren, der sowohl diskursstrukturierte Textpläne als auch vielfältige Texte enthält. Wir fanden heraus, dass unsere experimentelle Manipulation (die Sprecher zu bitten, sich für ihre Äußerungen verschiedene Zielgruppen vorzustellen) effektiv war, um Texte mit höherer und niedrigerer Informationsdichte hervorzurufen. Insbesondere schrieben unsere Teilnehmer Texte mit geringerer Informationsdichte, wenn sie sich vorstellten, sie würden einen älteren Angehörigen ansprechen. Wir stellten auch fest, dass die Teilnehmer die Paraphrasierungsaufgabe oft durch eine Neuordnung der im Originaltext präsentierten Informationen erledigten, worauf eingegangen sind, indem wir die zu den Originaltexten gehörenden Textpläne manuell korrigiert haben, um sie mit den von unseren Teilnehmern verfassten Texten abzugleichen. Dies führte zu einem Set von 1344 Texten mit unterschiedlicher Informationsdichte und Goldstandard-Anmerkungen zur Diskursstruktur. Dieses Korpus, das wir das „Extended SPaRKy Restaurant Corpus" (ESRC) nennen, ist das erste seiner Art, das die Unterschiede in der Ideendichte für kurze Texte widerspiegelt, die sich an ältere und jüngere Erwachsene richten.

Neben dem Sammeln von Daten für unsere Satzplanungsaufgabe, haben wir einen neuartigen Datensatz zur menschlichen Produktion von verweisenden Ausdrücken gesammelt (Kapitel 10). Dieses Korpus ist das erste deutsche Korpus, das dieselbe Art von Stimuli wie frühere zu verweisenden Ausdrücken u.a. im Englischen (van Deemter, van der Sluis & Gatt, 2006) und Niederländisch (Koolen & Krahmer, 2010). Darüber hinaus spiegelt es aufgrund unseres experimentellen Designs das menschliche Verhalten beim Sprechen mit Hörern unter kognitiver Belastung wider. Die in diesem Korpus vorhandene Variation stärkt unsere allgemeine Behauptung, dass NLG-Systeme in der Lage sein müssen, Variationen zu erzeugen, wenn wir natürliche Texte erzeugen wollen.

AUSWERTUNGEN FÜR GENERIERTEN TEXT

Wir haben in Kapitel 8 den Stand der Technik für automatische und menschliche Evaluierungen von NLG-Systemen untersucht, um die

beste Methode zur Evaluierung unseres Systems zu ermitteln. Unsere automatisierten Metriken konzentrierten sich auf die rohe Umfang möglicher Eingaben und identifizierten, welche Versionen unseres Systems LF für die meisten TP produzieren konnten und wie viele Texte wir als Ergebnis generieren konnten. Wir haben die automatische Metrik BLEU[4] auch als Textähnlichkeitsmaß verwendet, um zu beurteilen, inwieweit unterschiedliche Parametereinstellungen zu unterschiedlichen Texten führten.

Wir haben Skripte zur schnellen Bewertung der Textqualität durch Forscher und eine neuartige Crowdsourcing-Schnittstelle zur Bewertung durch Crowdworker entwickelt. Unser Schnellvergleichsskript hat sein Ziel erreicht, es einem Forscher zu ermöglichen, die relative Qualität verschiedener Texte schnell zu beurteilen und Vergleiche von 100 Textpaaren in nur 20 Minuten durchzuführen. Noch wichtiger ist, dass unsere Bewertungsschnittstelle für Crowdworker eine Möglichkeit bot, feinkörnige Bewertungen zu sammeln und die Bewertungen dennoch mit beschreibenden ‚Ankern' zu untermauern: durch die Verwendung einer gleitenden Skala und mit Textbeschreibungen entlang der Skala können die Teilnehmer ihre Bewertungen von ähnlicher Qualität besser differenzieren als mit mit einer einfachen 5-, 6- oder 7-stufigen Bewertungsskala, ohne den Mittelpunkt der Skala für etwas so Abstraktes wie „Fluency" schätzen zu müssen. Durch das Sammeln kontinuierlicher Daten konnten wir einfache parametrische statistische Tests anstelle der komplexeren Modelle verwenden, die für ordinale Daten erforderlich sind. Die Benutzeroberfläche lieferte auch Feedback zur semantischen Genauigkeit, obwohl die Teilnehmer mit dem Begriff „zusätzliche Details" zu kämpfen hatten und dies häufig als „enthielt das Output Fakten enthielt, die Sie für irrelevant halten?" inteprierten anstatt als „hat das System zusätzlich zu den oben gelisteten Fakten weitere Fakten ausgedrückt?". In der Umfrage wurde auch nach dem Ausdruck von Diskursbeziehungen gefragt; die Antworten auf diese Fragen waren jedoch weniger aufschlussreich, als sie hätten sein können.

Während unsere Teilnehmer offenbar ohne Hintergedanken teilgenommen und aufrichtige Anstrengungen unternommen haben, um unsere Fragen zu beantworten, scheint es für sie schwierig gewesen zu sein, zwischen Beurteilungen von „Fluency", Bewertungen der semantischen Genauigkeit und Wahrnehmungen von Diskursbeziehungen zu wechseln. Daher schlagen wir vor, dass sich künftige Evaluationsumfragen darauf konzentrieren sollten, jeweils nur eine oder zwei eng miteinander verbundene Fragen zu beantworten; sie können sich beispielsweise nur auf „Fluency", nur auf eingefügte und weggelassene Fakten oder nur auf die Diskursstruktur konzentrieren.

4 Aus dem Englischen „Bilingual Language Evaluation Understudy" (Papineni u. a., 2002)

ABSCHLUSS

Die in dieser Dissertation vorgestellte Arbeit konzentrierte sich auf das Erlernen von Satzplanungsregeln zur Generierung neuartiger Texte unter Verwendung synchroner Grammatiken. Wir definierten einen Formalismus zur Beschreibung dieser Generierungsregeln, sammelten einen neuartigen Datensatz für das Training diskursfähiger NLG-Systeme und implementierten und evaluierten ein solches System auf mehreren Datensätzen. Zusätzlich zu diesen praktischen Bemühungen untersuchten wir menschliche Variationen bei der Anpassung ihrer Äußerungen an Zuhörer unter kognitiver Belastung.

Diese Arbeiten zeigen, dass synchrone Grammatiken eine nützliche Repräsentation für Satzplanungsregeln darstellen, dass Bayessche nichtparametrische Modelle solche Grammatiken mit geeigneten Trainingsdaten induzieren können und dass solche gelernten Modelle bestehende neuronale neuronaler Netze in Bezug auf die semantische Treue übertreffen können. Allerdings eröffnet diese These eröffnet jedoch auch mehrere Richtungen für zukünftige Forschung, wie man die verschiedenen anspruchsvollen Aufgaben bei der Erzeugung natürlicher Sprache und wie man diese Systeme in Zukunft am besten evaluieren kann.

PUBLICATIONS

The following publications report on work presented in this thesis:

Howcroft, David M., Dietrich Klakow & Vera Demberg (Aug. 2017). "The Extended SPaRKy Restaurant Corpus: Designing a Corpus with Variable Information Density." eng. In: *Proc. of Interspeech 2017*. Stockholm, Sweden: ISCA, pp. 3757–3761. DOI: 10.21437/Interspeech.2017-1555. URL: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1555.html (visited on 06/05/2018).

Howcroft, David M, Dietrich Klakow & Vera Demberg (Nov. 2018). "Toward Bayesian Synchronous Tree Substitution Grammars for Sentence Planning." eng. In: *Proc. of the 11th International Conference on Natural Language Generation (INLG)*. Tilburg, the Netherlands: Association for Computational Linguistics.

Howcroft, David, Jorrig Vogels & Vera Demberg (2017). "G-TUNA: A Corpus of Referring Expressions in German, Including Duration Information." eng. In: *Proceedings of the 10th International Conference on Natural Language Generation*. Santiago de Compostela, Spain: Association for Computational Linguistics, pp. 149–153. DOI: 10.18653/v1/W17-3522. URL: http://aclweb.org/anthology/W17-3522 (visited on 08/22/2018).

Vogels, Jorrig, David M. Howcroft, Elli Tourtouri & Vera Demberg (2020). "How Speakers Adapt Object Descriptions to Listeners under Load." eng. In: *Language, Cognition and Neuroscience* 35.1, pp. 78–92. DOI: 10.1080/23273798.2019.1648839. URL: https://www.tandfonline.com/doi/full/10.1080/23273798.2019.1648839.

The following publications are related to the topics discussed in this thesis, but represent larger collaborations and are not reported in this thesis:

Belz, Anya, Simon Mille & David M Howcroft (Dec. 2020). "Disentangling the Properties of Human Evaluation Methods: A Classification System to Support Comparability, Meta-Evaluation and Reproducibility Testing." eng. In: *Proc. of the 13th International Conference on Natural Language Generation (INLG)*. Dublin, Ireland: Association for Computational Linguistics, pp. 183–194. URL: https://www.aclweb.org/anthology/2020.inlg-1.24/.

Demberg, Vera, Jörg Hoffmann, David M. Howcroft, Dietrich Klakow & Álvaro Torralba (2016). "Search Challenges in Natural Language Generation with Complex Optimization Objectives." In: *Künstliche Intelligenz* 30.1, pp. 63–69. DOI: 10.1007/s13218-015-0409-5. URL: http://dx.doi.org/10.1007/s13218-015-0409-5.

Dušek, Ondřej, David M. Howcroft & Verena Rieser (2019). "Semantic Noise Matters for Neural Natural Language Generation." eng. In: *Proc. of the 12th International Conference on Natural Language Generation (INLG)*. Tokyo, Japan: Association for Computational Linguistics. DOI: 10/ggwzgc.

Howcroft, David M, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood & Simon Mille (Dec. 2020). "Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions." eng. In: *Proc. of the 13th International Conference on Natural Language Generation (INLG)*. Dublin, Ireland: Association for Computational Linguistics, pp. 169–182. URL: https://www.aclweb.org/anthology/2020.inlg-1.23/.

Schwenger, Maximilian, Alvaro Torralba, Joerg Hoffmann, David M Howcroft & Vera Demberg (Dec. 2016). "From OpenCCG to AI Planning: Detecting Infeasible Edges in Sentence Generation." eng. In: *Proc. of the 26th International Conference on Computational Linguistics (COLING): Technical Papers*. Osaka, Japan: Association for Computational Linguistics, pp. 1524–1534.

ACKNOWLEDGMENTS

This thesis has been a long time coming and would not have been possible without many kinds of support from many kind people.

I want to begin by thanking Vera Demberg and Dietrich Klakow for serving as my supervisors and mentors, each providing their own unique perspective on research in general and my work in particular. Thank you both for supporting me when I doubted myself and for allowing me the freedom to follow my interests in carrying out this thesis work.

Vera and Dietrich also cultivated friendly and collegial lab environments. I'm grateful to have been a part of Friday croissants and bike-to-work challenges, group lunches in the mensa, new years' dinners, and outdoor excursions. Thank you Jorrig Vogels, Andrea Fischer, Katja Häuser, and Tony Xudong Hong for being good officemates. Special thanks, in no particular order, to Katja Kravtchenko, Elisabeth Rabs, Marc Schulder, Asad Sayeed, Alessandra Zarcone, Fatemeh Torabi Asr, Lisa Teunissen, Clayton Greenberg, Merel Scholman, Wei Shi, Jonas Groschwitz, Meaghan Fowlie, Aleks Piwowarek, Liesa Heuschkel, Leonie Lapp, Yoana Vergilova, Adrin Jalali, Ashkan Taslimi, Daniel McDonald, Mahsa Vafaie, Julia Dembowski, Margarita Ryzhova, Johannah O'Mahoney, Fraser Bowen, Esther van den Berg, and Te Rutherford. Thank you also to the many other members of LST and the SFB whose company made work fun.

A number of other researchers have influenced me in ways deserving thanks. Thank you to Leon Bergen for sharing your code and answering questions. Thanks to Michael White, Alexander Koller, and Ehud Reiter for influencing the way I think about natural language generation. Thanks to Detmar Meurers, Sina Zarrieß, and Anya Belz for influencing how I think about text quality and evaluations. Thank you to Verena Rieser and Dimitra Gkatzia for allowing me to continue to grow as a researcher in their labs in Scotland while I finished what I started in Saarland. Thank you to Ingo Reich for serving as my SFB mentor and to Elke Teich for your excitement for linguistics.

Thank you to the members of staff without whom this work would not be possible. Thanks to the IT team and sysadmins Bernd Mechenbier, Christoph Claudo, and Jochen Pätzold. Thanks to Angelika Obree, Gabi Reibold, and Claudia Verburg for administrative assistance and conversation. Thank you to Lena Steshenko, Marie-Ann Kühne, and Patricia Borrull Enguix for SFB-specific admin assistance. Thank you to Stefan Thater and Diana Steffen for keeping departmental admin running smoothly.

For feedback on various sections of this thesis during drafting, I must thank Vera and Dietrich, as well as Tony, Merel, Katja, Jonas, Elli, Sacha Beniamine, Aniello De Santo, and Loïc Grobol. Thank you Florian Pusse, Ben Peters, and Ahmed Sohail for your efforts as research assistants. And thanks to Stalin Varanasi, Jess Mankewitz, and Sacha for helping to test experimental interfaces.

I also want to thank the friends and family who helped me to remember there is more to life than a PhD—valuable perspective indeed! Thank you to Tobias Price for the years of love, support, and companionship. Thank you Mum, Dad, and Sam for always being there and believing in me. Thank you to Sacha Beniamine and Maya Wedin for reassuring me when I have despaired. Thank you Layla, Salem, and Cricket for being the best cats and Belinda for being the cattiest dog.

Thank you Saskia Riedel, Fran Whatever-You-Want, Laura Bitterlich, Lydia Rabkin, and Sabine Sur for being grand non-linguistic friends in Saarbrücken. Thank you to Jenny Singer, Katie Hood, and Amanda Cercas Curry for being voices of reason and always up for a stroll in Edinburgh. Thank you to the many friends who made conferences fun, including Emiel van Miltenburg, Jana Götze, Amy Isard, and Leo Lappänen. And thank you to the server.

As a final note of thanks, my PhD work was financially supported through SFB 1102 and Lehrstuhl Klakow.

I have almost certainly omitted someone very important and will remember them as soon as this is printed. If you are the person I have forgotten, please accept my sincere apologies and know that I am indeed grateful for your presence in my life.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

BLEU Bilingual Evaluation Understudy

CCG Combinatorial Categorial Grammar

CFG Context-Free Grammar

ConTRe Continuous Tracking and Reaction

CRP Chinese Restaurant Process

DAG Dependency Attachment Grammar

DFKI German Research Center for Artificial Intelligence

DOP Data-Oriented Parsing

DP Dirichlet Process

EB Experiment Builder

EM Expectation-Maximization

ESRC Extended SPaRKy Restaurant Corpus

GRU Gated Recurrent Unit

ICA Index of Cognitive Activity

ML Machine Learning
LCA least common ancestor
LF Logical Form
LM Language Model
LSTM Long Short-Term Memory
MT Machine Translation
MR Meaning Representation
NLG Natural Language Generation
NN Neural Network
POS Part of Speech
PSG Phrase Structure Grammar
PTB Penn Treebank
REG Referring Expression Generation
RST Rhetorical Structure Theory
RNN Recurrent Neural Network
SC-LSTM Semantically-Controlled Long Short-Term Memory
sDAG Synchronous Dependency Attachment Grammar
seq2seq sequence-to-sequence
SRC SPaRKy Restaurant Corpus
sTSG Synchronous Tree Substitution Grammar
TAG Tree Adjoining Grammar
TSG Tree Substitution Grammar
TP Text Plan
UD Universal Dependencies

1 INTRODUCTION

Whenever we want a computer to do something for us, we must program it to do so. So when we want a computer to generate a text, the question is: how do we program it to produce human language output? This thesis provides one answer to this question.

To begin with, it helps to specify what our inputs and our outputs to this Natural Language Generation (NLG) task look like. Let's use a common input representation, which we can build upon later:

```
Input:
  inform(name="John's Pizzeria", cuisine="Italian, Pizza",
         price="20", food_quality="very_good")

Possible Outputs:
  John's Pizzeria is an Italian, Pizza restaurant with
      very good food at 20 dollars a plate.
  There is an Italian, Pizza restaurant called John's
      Pizzeria which costs 20 dollars for very good food.
  John's Pizzeria has very good food quality and is an
      Italian, Pizza restaurant. Its price is 20 dollars.
```

Figure 1: Example input with several possible outputs for a NLG system.

Here we have a collection of attributes and their values which we want a system to `inform` users about, along with three possible realisations of this input.

We can imagine a few approaches. First, having played madlibs[1] or used mail-merge[2] and wanting to keep things simple, we could specify a few templates, as in Figure 2. Right away we notice a few things, however. These templates don't actually work with the texts that inspired them, because of the a/an alternation in English. So we will need to implement some kinds of rules (or a proliferation of templates) to capture this regularity. Also note that each of these templates assumes all four attributes are specified: `name`, `cuisine`, `price`, and `food_quality`. This means that we either need to create templates for all possible combinations of attributes *or* create rules for how to combine (parts of) templates so that these combinations can be handled programmatically.

1 https://en.wikipedia.org/wiki/Mad_Libs
2 https://en.wikipedia.org/wiki/Mail_merge

```
Templates:
  <NAME> is a <CUISINE> restaurant with <FOOD_QUALITY>
      food at <PRICE> dollars a plate.

  There is a <CUISINE> restaurant called <NAME> which
      costs <PRICE> dollars for <FOOD_QUALITY> food.

  <NAME> has <FOOD_QUALITY> food quality and is a
      <CUISINE> restaurant. Its price is <PRICE> dollars.
```

Figure 2: Possible templates for the texts in Figure 1.

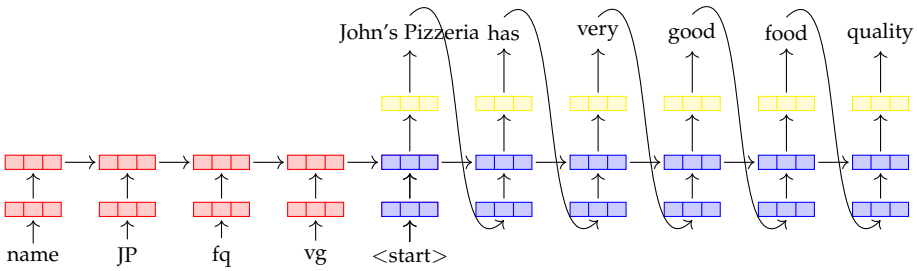


Figure 3: A simple neural model for NLG using an *encoder* (red rectangles) and a *decoder* (blue rectangles) to map inputs to outputs.

Writing such rules for transforming input into output would certainly feel a lot more like programming, but it also sounds like a lot of work; wouldn't it be nice if we could give the computer a collection of inputs and outputs and have it 'just figure it out' for itself?

Such Machine Learning (ML) approaches to natural language generation have become increasingly common over the past twenty years, especially with the dramatic improvements in compute power and tooling which have made Neural Network (NN) approaches to ML more feasible. Figure 3 shows what one such approach looks like.

This approach is a so-called *sequence-to-sequence* model, which does not restrict the input only to specific sets of attributes anticipated by the programmers and does not require explicit rules in order to generate text. Each rectangle in this figure represents a real-valued vector while the arrows represent matrix operations for creating one vector from others. The *encoder* (red rectangles) receives the input as a sequence of tokens and creates a single 'hidden' vector (the red rectangle with an arrow connecting it to a blue rectangle) to represent this input. The *decoder* (blue rectangles) receive the current 'hidden'

representation of the input as well as the preceding output token.[3] Each of the hidden states of the decoder is mapped to an output token through the yellow output nodes.

In this approach we can avoid manually constructing rules which transform inputs into natural language outputs. However, the matrices which our machine learning algorithms must learn are difficult to interpret, which makes debugging the resulting NLG system challenging. Moreover, current algorithms for fitting such neural network models require many input-output pairs in order to fit the model and often struggle with accuracy, failing to convey information present in the input or inserting information not given in the input.

We explore in this thesis whether it is possible to get the best of both worlds: automatically learning to generate while maintaining interpretability, by using machine learning to acquire rules for a rule-based system. Leveraging ML allows us to save the expert time and attention that would otherwise need to be spent writing rules, while making rules the target of our ML ensures that the resulting system is more easily interpreted and corrected. This is the approach we advocate for in this thesis.

In particular, we set out to understand the limits of Bayesian nonparametric methods for inducing synchronous grammars which can be used to generate varied texts. This raises two primary **research questions**:

1. Can we build a system which leverages Bayesian nonparametric methods to learn synchronous grammars for sentence planning?

2. Can such a system learn to produce *varied* texts given appropriate training data?

However, these questions need to be broken down in order to be approachable. To answer (1), we must begin by asking: (1a) what does a synchronous grammar which is appropriate for sentence planning look like? Given such a grammatical framework, we need to know: (1b) how do existing approaches to inducing synchronous grammars need to be adapted in order to model the kind of grammar we would like to generate? Then we can implement and test such a model to ultimately answer (1).

For (2), we must first assess whether any datasets exist which are appropriate to learn the kind of variation we are interested in. For the purposes of this thesis we will focus on variation with respect to *information density* (also known as *surprisal*; Shannon & Weaver, 1948). Therefore we explore (2a) how to adapt crowdsourcing methods to collect a corpus of texts which vary with respect to information density Given such a corpus, we can then ask (2b) whether our model

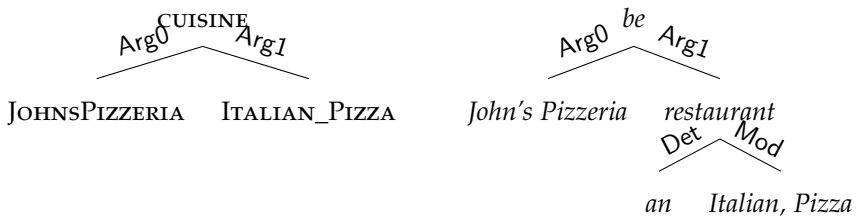
3 The first hidden state in the decoder receives a 'start' token since there is no preceding output token.

learns to reproduce this variation appropriately. Along the way we can also explore (2′): why is variation desirable in NLG systems?

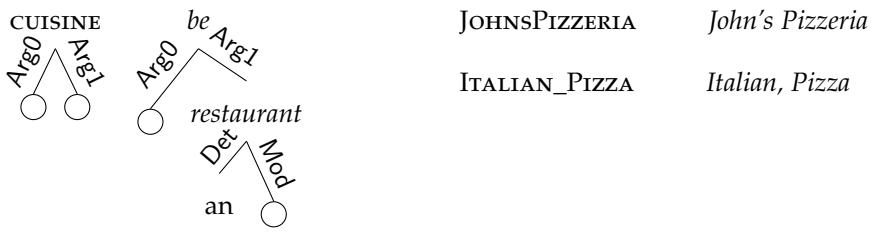
The next few pages lay out our approach to these questions at a high level, after which we describe the particular contributions of this thesis.

1.1 HOW CAN WE REPRESENT RULES?

To see how we might approach question (1a), let's consider a small example of rule-based NLG for part of the text shown above, *John's Pizzeria is an Italian, Pizza restaurant*. We have an input representation (semantic tree; left) and an output representation (syntactic tree; right). Note that this approach assumes we have a system in place to read off a text from a syntactic tree and that the rules are focused on mapping from (pseudo-)semantic inputs to (pseudo-)syntactic outputs.



The NLG system, then, must be able to map the tree on the left to the tree on the right. This could be done trivially, by storing a rule which produces the right-hand tree whenever it encounters the left-hand tree. However, such an approach misses an obvious generalization: if we need to describe the CUISINE offered at a variety of restaurants, then it makes sense to factor out the semantic arguments and their corresponding syntactic realizations. A set of rules like the following would therefore be re-usable in more circumstances:



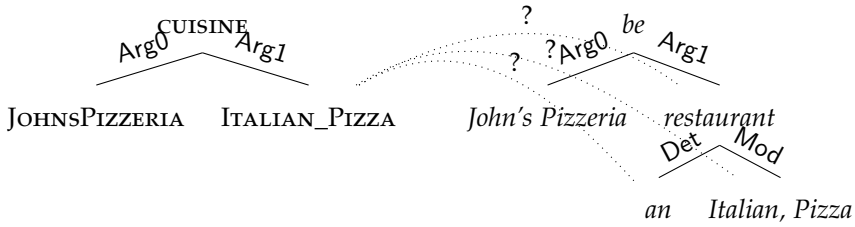
We can read the first of these rules as saying, 'when encountering a CUISINE node on the left-hand side, we can create a particular tree rooted at *be*, map the semantic Arg0 to the syntactic Arg0 using another rule, and map the semantic Arg1 to the syntactic Mod using a third rule'.

These kinds of rules, where we have the same number of unfilled arguments on the left and right sides of the rule, are well-described by Synchronous Tree Substitution Grammars (sTSGs). In Chapter 3 we

develop the argument that such rules are useful for NLG and propose an extension to make them an even better fit.

1.2 HOW CAN WE LEARN RULES?

In order to acquire sTSG rules, we will need to figure out how we can split pairs of semantic and syntactic trees into smaller pieces. Building on our running example, how do we know which of these possible alignments for ITALIAN_PIZZA is correct?



Chapter 5 describes existing approaches to grammar induction for stand-alone Tree Substitution Grammars (TSGs) as well as sTSGs. In short, we define a statistical model for what the semantic trees and syntactic trees can look like and how they can relate to each other before fitting that model on the basis of some training data. This provides the basis for our answer to (1b), which we explore with extensions and implementations described in Chapters 6 & 9.

1.3 WHAT DATA CAN WE USE?

In this thesis we focus on two kinds of data: datasets based on existing NLG systems and datasets which incorporate variation with respect to *information density*. Existing systems provide a reasonable starting point for testing a new approach to ML for NLG: we have access to a large number of possible valid outputs along with their inputs and we can ensure that our new approach is able to achieve similar performance compared to an existing system.

However, these datasets will also be necessarily constrained: they meet the information needs of the original developers and do not necessarily represent a natural way of speaking for many users.[4] In Chapter 7 we describe the desiderata our training data along with a survey of existing datasets before describing a novel means of collecting a training corpus with varied information density and presenting such a corpus that we have collected.

4 See, for example, *Its price is 20 dollars* describing the average price of a meal at a restaurant, per our running example.

1.4 HOW DO WE EVALUATE?

With training data and models in hand, we can generate novel texts, but how do we know if we are doing a good job? On the one hand, automated metrics are extremely reproducible and can provide consistent scores for a given text. On the other hand, automated metrics are often based on unrealistic assumptions or serve as poor proxies for actual human judgements of text quality. Human evaluations provide a kind of 'gold standard' for evaluation, relying on humans to directly assess the quality of a text.

Chapter 8 describes common approaches to both automated and human evaluations along with a survey of important features to consider in designing a novel evaluation method. Ultimately, this serves as the basis for a new human evaluation interface, also described in this chapter, which is then used in Chapter 9 to evaluate our implemented system.

1.5 WHY ARE WE INTERESTED IN VARIATION?

There has been some sleight of hand so far in this introduction. Our focus is primarily on ML for NLG, so we have focused on the infrastructure necessary for this task: a formalism, a machine learning method, datasets for training, and an approach to evaluation. However, we have so far assumed that, beyond merely training an NLG system, it is desirable to develop systems which can produce *varied* outputs.

Therefore we also collaborated to develop a psycholinguistic experiment focusing on variability in the context of *referring expression generation* produced by humans. When speakers produce a natural language utterance, they must decide how to refer to each of the entities they want to mention. For example, we might refer to the man pictured in Figure 4 as 'the grey-haired man in a suit', 'the former leader of my country of birth', or 'Nelson Mandela', depending on the context of our utterance.

In the work described in Chapter 10, we explore how speakers adapt their referring expressions when their listeners are under cognitive load, finding considerable variability depending on context. This contributes to an answer for (2′): an NLG system should produce variability if it is intended to produce human-like utterances.

1.6 CONTRIBUTIONS

The main contributions of this thesis include:

1. Formulation of Sentence Planning rules in terms of synchronous Tree Substitution Grammars (1a)

Figure 4: A picture of Nelson Mandela taken during his first trip to the USA in 1994. Copyright John Matthew Smith, 2001.

2. Formalization of Dependency Attachment Grammars as an extension to TSGs and extension to the synchronous case (1a)

3. Implementation of a Bayesian nonparametric model and Gibbs sampler for learning sTSGs for sentence planning (1b)

4. Development of the Extended SPaRKy Restaurant Corpus with varied information density and a rich semantic representation (2a)

5. An evaluation method for assessing linguistic fluency, semantic accuracy, and expression of discourse relations (1b)

6. Development of a corpus for Referring Expression Generation to explore human variation and its relevance for NLG (2′)

Contributions 1 and 2 relate to Question (1a), as we shall see in Chapter 3. Contribution 3 addresses Question (1b) with background provided in Chapters 4 & 5 and details given in Chapters 6 & 9.

Contribution 4 relates to Question (2a), addressed in Chapter 7, while Contribution 5 relates to Question (2b), which is addressed in Chapter 8. The final contribution (Contribution 6) stands apart from the rest, providing some context for understanding variation within and between human speakers to motivate our interest in variation for NLG.

Part I

BACKGROUND

To understand this work, you need to know something about:

- natural language generation,
- synchronous grammars,
- Chinese Restaurant Processes & Gibbs sampling, and
- previous work on Bayesian grammar induction.

2 NATURAL LANGUAGE GENERATION

Natural Language Generation (NLG) is an area of natural language processing which focuses on the task of transforming non-linguistic information into a natural language text. Approaches to NLG fall into two broad categories: some version of the modular, pipeline approach described in Reiter & Dale (2000), which we will call the *traditional NLG architecture*; and the end-to-end Machine Learning (ML) approach found in what we call *end-to-end systems*.

traditional NLG architecture
end-to-end systems

This chapter presents the traditional NLG architecture before discussing trends in sentence planning research in particular. It also provides an overview of current end-to-end systems.

The background on sentence planning provides context for our discussion of how to use synchronous tree-adjoining grammars for this purpose in the next chapter. The general overview and focus on sentence planning will help the reader to understand the theoretical framework we propose in Chapter 6, while the discussion of end-to-end alternatives motivates the baseline model we use in Chapter 9.

2.1 TRADITIONAL NLG ARCHITECTURE

The traditional NLG architecture divides the process of transforming non-linguistic information into a natural language text into three parts: document planning, sentence planning, and surface realization. In this section, we sketch each of these areas, roughly following Reiter & Dale (2000).

2.1.1 *Document Planning*

Document planning, also known as text planning or macroplanning, involves both content selection and high-level document structuring. While it is often convenient for NLG researchers to assume that content selection—the choice of what to say—is 'beyond the scope' of their task[1], industrial settings demand the delivery of a usable—and useful—system. This requires developers to pay at least as much attention to content selection as to the actual process of converting that content into text, despite the fact that the latter is often the more interesting task for a linguist.

This problem decomposition, into content selection and everything else, is often referred to with a two phrase summary of the NLG problem: our task is trying to figure out both *what* to say and *how* to

1 and, indeed, we do the same

Table 1: A collection of facts in the restaurant domain.

Restaurant	Cuisine	Price	Food Quality	Service	Decor
John's Pizzeria	Italian, Pizza	20	very good	very good	good
Caffe Buon Gusto	Italian	26	good	very good	very good

```
1. cuisine(JP, "Italian, Pizza")
2. cuisine(CBG, Italian)
3. price(JP, 20)
4. price(CBG, 26)
5. food_quality(JP, very_good)
6. food_quality(CBG, good)

contrast(1,2)
contrast(3,4)
contrast(5,6)
```

Figure 5: One collection of relevant facts from Table 1 and a set of CONTRAST relations between pairs of these relevant facts.

strategic and tactical generation

say it. These two aspects are also referred to as *strategic and tactical generation*, respectively.The remainder of this subsection, and the following subsections, focuses on determining *how* we should express some content once it has been chosen.

The first step in answering this question is to determine high-level document structure. This can range in complexity from simply choosing an order in which to express the chosen facts to defining the discourse structure over these facts, highlighting specific discourse relations in the process.

Consider, for example, the constellation of facts presented in Table 1. Given these facts and a set of user preferences, a content selection system must decide which are relevant for the user. If the system decides, then, that the cuisine, price, and food quality are the most relevant facts to communicate and that these two restaurants need to be compared to each other, the system will identify that these two restaurants differ along all of these dimensions and may choose to highlight this difference explicitly with the rhetorical relation CONTRAST. The resulting *text plan* (or document plan), then, will at least contain a collection of these six facts along with markers for the contrasts to highlight, as in Figure 5.

Depending on where the developers draw the line between document planning and sentence planning, these relations can be further constrained, indicating a hierarchical text structure, as in Figure 6.

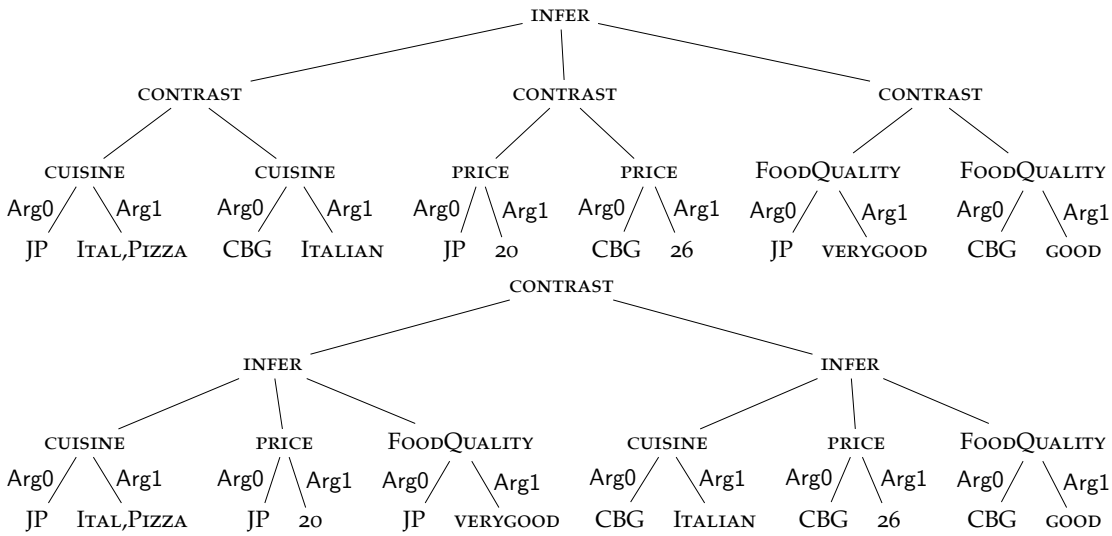


Figure 6: Two possible textplans encoding specific strategies for expressing the contrasts shown in Figure 5. For discussion of these different strategies and an explanation of the INFER relation, see Section 7.3.2.1.

Note the two possible document structures: the 'back-and-forth' text plan which talks about each property in turn, explicitly contrasting the two restaurants along that dimension; and the 'serial' text plan which describes each restaurant in its entirety and highlights the contrast only at the macro level.[2]

These trees can continue to be treated as unordered or the document planning system can fix the order of presentation as part of the discourse structure. In the sentence planning systems trained for this thesis (Ch. 9),we assume that the order is fixed by the document planner and use ordered trees of this kind as the input to the sentence planner.

2.1.2 *Sentence Planning*

Sentence planning, or microplanning, is the process of converting a document plan into a morphosyntactically specified representation which can be used as input to a surface realizer. Sentence planning typically includes lexicalization, aggregation, and referring expression generation. These are usually listed as separate components, although it can be argued that referring expression generation is a special case of lexicalization relating to entities and that aggregation interacts with lexicalization.

2 The interested reader can learn more about these two document structures in Nakatsu & White (2010).

Lexicalization is basically lemma selection, deciding which natural language words should be used to express the contents of the document plan This means choosing:

- individual words and phrases (e.g., 'very good' vs. 'great')
- possible syntactic roles for semantic arguments (e.g., whether the semantic agent should be the syntactic subject or the syntactic object)
- whether an argument should be expressed as a noun phrase or a prepositional phrase
- nominalization of properties verus predication of properties (e.g., 'good decor' vs. 'well-decorated')

Closely related to this, referring expression generation (REG) addresses the specific problem of choosing adequate expressions for referring to individual entities in the discourse.[3] In the context of generating restaurant recommendations, for example, we might refer to SONIA ROSE by name, by pronoun, or by a noun phrase headed by 'restaurant', as in 'This Italian restaurant'. When choosing to use a pronoun or other less explicit referring expression, the system must be aware of the context in which this expression is being generated. This is why referring expression generation is usually considered *after* other forms of lexicalization: for example, referring to a restaurant pronominally during its first mention can lead to ill-formed or confusing texts, even if the intended referent is cataphorically recoverable.[4]

Aggregation also depends on lexicalization choices, although some amount of semantic aggregation can already be performed during document planning, as is illustrated in the choice to group properties of a single restaurant together in the 'serial' text plan exemplified in the previous section. To borrow from the *decor* example highlighted above, if we have to express DECOR(SONIAROSE, GOOD) and FOOD_QUALITY(SONIAROSE, GOOD), our aggregation options depend in part on syntactic restrictions imposed by our choice of lexicalization. If we say that *Sonia Rose has good decor*, then we can say that *Sonia rose has good decor and good food*. If, however, we say that *Sonia Rose is well-decorated*, then we must at least realize two separate verb phrases (e.g. saying that *Sonia Rose is well-decorated and (it) has good food*).

3 Note that REG can also depend on world knowledge, leading some to blur the line between document planning and sentence planning (as with SPUD, discussed in Section 2.2.1).

4 'It has good decor and Sonia Rose is cheap' sounds like 'It' and 'Sonia Rose' refer to different entities. Consider, however, 'With its low prices and delicious food, Sonia Rose is a bargain' or 'It has great food at amazing prices! Come to Sonia Rose today!'.

2.1.3 *Surface Realization*

Surface realization then completes the process of natural language generation by ensuring that the lexical items and other choices made during sentence planning are transformed into a grammatical text in the target language. This transformation requires putting words into the correct order (*linearization*) and handling morphosyntactic agreement (e.g., *I like* instead of *I likes*, *I to like*, or *me like*). Issues of correct capitalization and punctuation can then be handled as a part of the surface realizer proper or as a post-process.

linearization

2.2 RESEARCH ON SENTENCE PLANNING

The need for an intermediate representation between the document planner and the surface realizer was recognized by Meteer (1990). Prior work in generation had focused on a coarse expressive ability at the level of individual clauses, which ensured that every proposition could be expressed but failed to utilize the full power of natural language. For example, since surface realization resources were only guaranteed to exist at the level of a single proposition being expressed as a whole clause, these systems often had to resort to conjunctions of simple clauses instead of encoding meaning in "complex noun phrases, nominalizations, adverbial phrases, and other adjuncts". Moreover, these systems could not account for possible interactions depending on which linguistic structures were chosen to express a particular proposition.

Over the last twenty years, research in sentence planning has primarily fallen into two groups. The first line of research blurs the line between document planning, sentence planning, and surface realization, using techniques from *AI planning*. The second line of research is more similar to our own approach, using an explicit sentence planning stage within the traditional pipeline architecture.

AI planning*: a field of artificial intelligence which researches general search techniques which can be applied to a wide range of problems*

2.2.1 *Extended Sentence Planning*

The first group follows SPUD[5] (Stone & Doran, 1997; Stone et al., 2003) in approaching sentence generation from a declarative perspective and combining document planning, sentence planning, and surface realization into a joint problem. Stone & Doran (1997) begin by casting all of NLG as a sort of referring expression generation: the objective is to generate *descriptions* which distinguish an intended meaning from possible distractors in the context of some world knowledge. The input to the algorithm is a flat semantics which is checked against a similarly encoded knowledge base to see what if any additional propositions need to be mentioned for a complete distinguishing de-

5 Sentence Planning using Descriptions

scription. This means that, while there is still some amount of content selection happening before the input is passed to SPUD, the system augments this meaning representation based on its world knowledge. Realization in this approach associates pieces of this semantic representation with lexicalized Tree Adjoining Grammar (TAG) trees and uses standard combination operators to generate output.

Contemporaneously with SPUD, Marcu (1997) described the computational difficulty in determining how to use the available syntactico-semantic rules for combining propositions and building a text. This led Mellish et al. (1998) to explore a variety of stochastic approaches to navigate this large search space. Koller & Stone (2007) recast the problem in the language of the AI planning community, making it possible to use off-the-shelf systems for heuristic search to do sentence planning and surface realization, which Koller & Hoffmann (2010) then continued.

In addition to work on improving the efficiency of this approach to generation, others explored incorporating features to allow the system to align its lexical choices with those of an interlocutor (Buschmeier, Bergmann & Kopp, 2009).

2.2.2 *Explicit Sentence Planning*

The second group follows Walker, Rambow & Rogati (2001) in using hand-crafted rules for sentence planning and splitting the task in two, first overgenerating possible sentence plans and then re-ranking them based on a trained model of user preferences.

Walker, Rambow & Rogati (2001) used a set of hand-crafted rules to over-generate information-seeking texts for a dialogue system in the travel planning domain. Stent, Prasad & Walker (2004) extended this system, showing that it could be applied to another domain (restaurant recommendations) and that it scaled up to handling texts with a more complex discourse structure. Walker et al. (2007) used this trainable approach to sentence planning to learn not just general preferences for overall text quality, but to adapt sentence planning choices to individual users.

All of these studies used hand-crafted rules for sentence planning, designed for a limited domain. Stent & Molina (2009) aimed to extract domain-independent sentence planning rules from the Penn Treebank (PTB) using PropBank (Carlson, Marcu & Okurowski, 2001) and the Rhetorical Structure Theory (RST) discourse treebank (Palmer, Gildea & Kingsbury, 2005, RST-DT). Their extracted yielded rules for 53 of the 57 core relations in the RST-DT with coverage for up to 205 *discourse cues*. The full set of sentence planning patterns consists of 5810 partially specified plans, which they offer to the research community. There is no human evaluation of the quality of texts produced using these rules.

discourse cues: *words introducing a specific discourse relation*

Lukin, Reed & Walker (2015) explores variation in sentence planning applied to a storytelling domain. They adapt an earlier NLG system for fables to the domain of personal narratives extracted from weblogs. This corpus of personal narratives includes annotations for discourse relations which are encoded by the original NLG system into the input format for their surface realizer. By 'de-aggregating' these surface realizer inputs, they are able to 're-aggregate' the texts by applying different sentence planning rules. This approach allows them to generate variations of the same stories in their dataset. Their extended system was able to produce variations which outperformed a purpose-built story-telling NLG system, but did not perform as well as the human written texts.

As with all of the systems in this second line of research, our work assumes that we have tree structured text plans as input to our sentence planner. Where Stent & Molina (2009) focused on span ordering, sentence aggregation, and discourse cue selection, we also learn lexicalization rules. As with Lukin, Reed & Walker (2015), we are interested in generating variations on the same text plans, although our system learns the required sentence planning rules.

2.3 END-TO-END SYSTEMS

While we have established that traditional systems are not necessarily deterministic, they do generally make use of hand-crafted rules for each stage of the pipeline. This requires a substantial investment of human time and attention, both for initial development as well as for porting an existing system to new domains.

Of course the idea of reducing the required engineering effort is appealing, so there have been efforts to eliminate it entirely by using machine learning to automatically train *end-to-end* NLG systems. The idea in these approaches is to collect a corpus of in-domain texts accompanied by some semantic representation (often the CUED dialogue act scheme (Young, 2009)) and then apply, e.g., Neural Network (NN)-based approaches to learn how to transform the input, semantic representation into text directly.

The idea of using data-driven methods to reduce the effort of building NLG systems is not new. About 20 years ago several groups proposed using statistical approaches to NLG (Knight & Hatzivassiloglou, 1995; Oberlander & Brew, 2000; Oh & Rudnicky, 2002; Rambow, Bangalore & Walker, 2001). These systems demonstrated the feasibilty of the *overgenerate-and-rank* paradigm of NLG, using a (cascade of) statistical model(s) to generate reasonably fluent alternatives for a particular meaning and then re-ranking the n-best alternatives to select the system output. These models typically preserved at least part of the traditional pipeline to structure the generation problem and did not see widespread adoption.

overgenerate-and-rank

MR	`inform(name="Ali Baba", type=placetoeat, eattype=restaurant, area=riverside, near="The Bakers", near="Avalon")`
text	*Close to both the Bakers and Avalon you will find the riverside restaurant, The Ali Baba*
MR	`inform(name="Lan Hong House", food=Chinese, eattype=restaurant, area=riverside)`
text	*A Chinese restaurant alternative for riverside dining is the Lan Hong House*

Table 2: The first two meaning representations (MRs) and texts from the BAGEL corpus (Mairesse et al., 2010). Further details of the corpus given in Section 7.4.2.

The next section describes BAGEL, one of the first attempts to minimize reliance on the traditional NLG pipeline and develop a more fully end-to-end approach to automatically learning NLG systems. After that we summarize some of the prominent neural-network approaches arising in recent years and touch on the end-to-end generation challenge.

2.3.1 *Statistical Methods*

Mairesse et al. (2010) developed the BAGEL system[6], to generate dialogue system responses in the restaurant recommendation domain. To train their system they collected a dataset consisting of 202 sets of slot-value pairs (their input meaning representations) paired with texts. Some examples are pictured in Table 2.

Mairesse et al. (2010) first split these MRs into a sequence of 'mandatory semantic stacks' (e.g., `inform(name(Ali Baba))`, `inform(type(placetoeat))`, `inform(eattype(restaurant))`, etc) and introduce the notion of an 'intermediary semantic stack', which represents only partial semantic information. Looking at the first example in Table 2, the words *Close to both the ... and ...* would be annotated with `inform(near)`, because they contribute toward the expression of this value although they do not realize the slots' values themselves. The words *Bakers* and *Avalon* would have the annotations `inform(near(The Bakers))` and `inform(near(Avalon))`, respectively.

Their end-to-end model, then, is trained on sequences of semantic stacks aligned through such annotations to the texts in their corpus. This dynamic Bayesian network first orders the mandatory semantic stacks and then inserts intermediary semantic stacks before sampling a probable sequence of words based on the final sequence of semantic stacks.

6 further detailed in Mairesse & Young, 2014

They found that their model, trained using active learning, performed nearly as well as human-written texts according to human subjects (naturalness score of 4.0 v. 4.07 and informativeness score of 4.07 v. 4.13, both scores on a 5-point scale). While their model was the first end-to-end system trained to generate text directly from an input meaning representation, the reliance on initial alignments led to difficulties in scaling the system. These difficulties motivated later neural methods (see Sec. 2.3.2) to focus on learning to generate from unaligned MR-text pairs.

A recent approach similar in spirit but simpler than the BAGEL system comes from Mahapatra, Naskar & Bandyopadhyay (2016). For their task Mahapatra et al. focused on learning to generate weather forecast texts based on tuples of non-linguistic weather data.By matching non-linguistic (e.g. numeric) strings in the corpus texts, their system learns the order in which these values should be expressed. Rather than assigning partial meaning to the phrases occurring between these values (as in BAGEL's intermediary semantic stacks), Mahapatra et al. extract these so-called 'interlinked word groups'. Their approach, then, predicts the sequence of values which must be mentioned, and then directly predicts what word sequences should occur before, after, and between these values. This very simple approach allowed their system to perform comparably to several others which required more human effort on this highly constrained domain.

These systems illustrate that focused end-to-end solutions can work for highly constrained domains even with limited data, but the difficulty in extending these systems and getting them to generalize contributed to interest in the neural models which we discuss next.

2.3.2 *Neural Methods*

The work of Oberlander & Brew (2000), Rambow, Bangalore & Walker (2001), Oh & Rudnicky (2002), and Ratnaparkhi (2002) cast stochastic NLG as sampling from language models, often including some amount of *delexicalization*. These works are kindred spirits to the recent wave of neural NLG models.

delexicalization: *replacing tokens with their semantic class or* slot

Wen et al. (2015b) presented one of the earliest end-to-end models for neural NLG, combining a Recurrent Neural Network (RNN) Language Model (LM) with the input meaning reintroduced at each word of the sampling process. This approach used several re-ranking techniques to try to improve semantic coverage, but the Semantically-Controlled Long Short-Term Memory (SC-LSTM) proposed in (Wen et al., 2015a) provided finer control by introducing the meaning representation only once and using a learned gating mechanism to decide when to generate which portions of the meaning.

Dušek & Jurčíček (2016) took an alternative approach, modelling the task as a sequence-to-sequence (seq2seq) problem in their TGen

system. In this approach, the meaning representations of (Wen et al., 2015a; Wen et al., 2015b) are decomposed into individual slot-value pairs which are then presented sequentially to an Long Short-Term Memory (LSTM) encoder. This approach allowed their system to learn from less training data than the SC-LSTM.

More relevantly to the current thesis, this work also explored learning to generate syntactic representations which could be used with an existing surface realization system. Unfortunately, they did not conduct a human evaluation, so it is unclear whether the slightly lower performance of their syntax-based system with respect to BLEU scores corresponded to lower quality texts overall.

Kiddon, Zettlemoyer & Choi (2016) presented yet another architecture for neural NLG, called the *neural checklist model*. For their RNN LM they use a Gated Recurrent Unit (GRU) with a novel attention mechanism designed to keep track of which components of the input semantics have already been expressed and which have not. Their target task was recipe generation, where longer texts should mention certain 'semantic' entries multiple times (e.g. mentioning an ingredient at various stages of processing in the course of the recipe) and all items need to be mentioned at least once. They found that their system performed better with respect to mentioning the intended ingredients in this task, realizing 83% of the listed ingredients and adding less than one extra ingredient on average. For comparison with other systems, they also applied their system to two of the dialogue-system-oriented tasks used by Wen et al. (2015a). On automated metrics (BLEU), they found that their system outperformed the SC-LSTM (91 v. 87 in the hotel domain; 78 v. 75 in the restaurant domain).

What the approaches discussed so far have in common is a relatively flat semantic structure: they use 'meaning representations' which consist of sets of pairs of slots and their values. This means that they effectively focus only on expressing these collections of facts, without any effort to structure the information at a higher level. Recent work has begun to introduce these aspects of document and sentence planning into neural NLG.

Nayak et al. (2017) explored two approaches to subdividing the generation task on a sentence-by-sentence basis. In their FLAT approach, they simply split the meaning representations and texts at the sentence level and realized each sentence separately. In their POSITIONAL approach, they did the same but incorporated an initial token to indicate whether the sentence being generated was at the *beginning* of a text or occurred *inside* the text. This latter approach significantly outperformed their baseline, which did not include any 'sentence planning', in a human evaluation (3.0 v. 2.8 grammaticality score; 2.8 v. 2.7 overall score; both on a 5 point scale).

Reed, Oraby & Walker (2018) also examined sentence scoping, although they used an additional token to represent the target text

length in sentences to do so rather than splitting the sets of slot-value pairs explicitly. They also explored single-token indicators in the meaning representation as a way of guiding aggregation of similar values across slots and expression of the discourse relation CONTRAST.

Because they were interested in these specific phenomena and had good automated metrics for these objectives, they did not conduct a human evaluation. However, their automated evaluations suggest that the single indicator token approach to augmenting meaning representations with sentence planning objectives is promising: these initial tests achieved greater than 80% accuracy on each of their tasks with slot-accuracies ranging from 3 to 24% across all three of their tasks.

In this section we have summarized just a few notable approaches to neural NLG, but we would be remiss if we did not also mention the end-to-end generation challenge[7]. This challenge ran throughout 2017 with the results being presented at the International Conference on Natural Language Generation in 2018.

For this challenge, subjects had access to a corpus of 50k texts and meaning representations to either train their systems or use as guidance in developing a traditional system. The majority of the primary systems (12/20) submitted to the challenge used a variant on the seq2seq approach, with two more systems using variants of the SC-LSTM approach. One non-neural data-driven system, two rule-based systems, and three template-based systems were submitted. These statistics are roughly representative of the distribution of papers being published on novel natural language generation systems today.

2.4 CONCLUSION

We have seen the traditional subdivision of natural language generation into document planning, sentence planning, and surface realization. This thesis focuses on the second of these areas, developing an approach for automatically learning sentence planning rules. This approach allows us to operate within the traditional, rule-based architecture while also leveraging machine learning to facilitate system development.

We also presented recent work in end-to-end NLG which aims to do away with the pipeline as much as possible and directly learn to generate natural language outputs for simple input meaning representations. These approaches so far struggle with semantic completeness and do not have a higher level notion of text structure at the discourse level. In applying machine learning for sentence planning, this thesis addresses these specific weaknesses.

7 http://www.macs.hw.ac.uk/InteractionLab/E2E/

With this background in mind, we can now shift to understanding how sentence planning rules can be expressed using synchronous grammars (Ch. 3) and what methods we can use for inducing such grammars (Ch.s 4 & 5).

3 SYNCHRONOUS GRAMMARS FOR SENTENCE PLANNING

This chapter introduces the grammatical formalisms we will use to represent sentence planning rules. We begin by introducing Tree Substitution Grammars (TSGs) before describing their synchronous variant and examining the strengths and weaknesses of Synchronous Tree Substitution Grammars (sTSGs) as a representation for sentence planning rules. This discussion motivates the introduction of Dependency Attachment Grammars (DAGs), for which we provide the first formal definition and an extension to cover synchronous derivations. We conclude by revisiting our discussion of sentence planning rules in light of this new formalism.

The models implemented in Chapter 9 are based on the sTSGs presented here.

3.1 TREE-SUBSTITUTION GRAMMARS

We are interested in building trees. One way to do this is with tree substitution grammars.

A tree substitution grammar consists of a set of *elementary trees* which can be used to expand *frontier nodes*, beginning with a root node, until there are no frontier nodes left and we have a complete tree. But let's look at some example trees to make this all more concrete.

elementary trees

frontier nodes

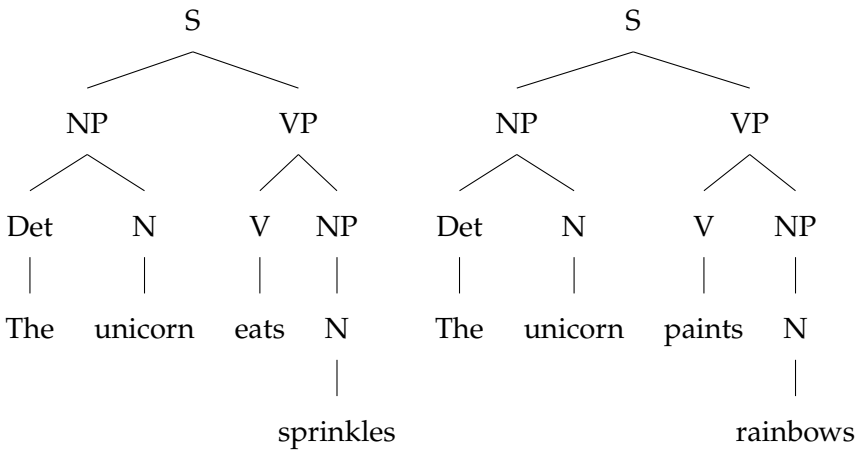
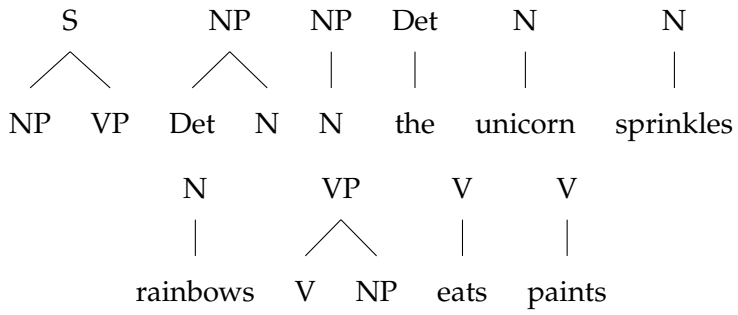
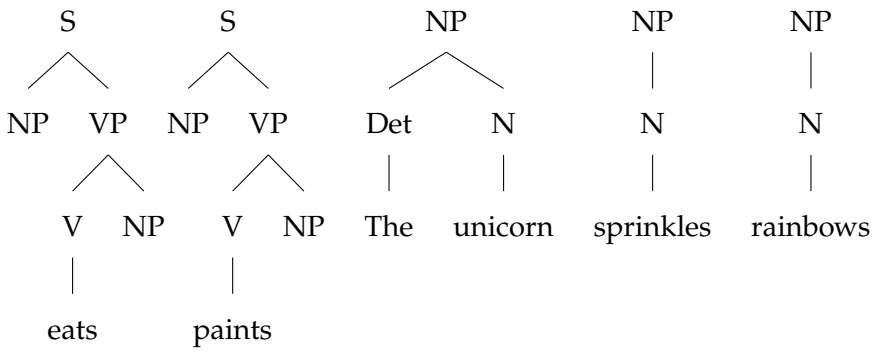


Figure 7 shows a few possible elementary trees we could use to derive these trees. On the one hand, we could have the set of trees in (a), which correspond to normal context-free rules for deriving a



(a) CFG-like rules



(b) 'Ideal' TSG rules.

Figure 7: Sample elementary trees for the examples above.

phrase-structure tree. If we only have rules like this, however, we will need many derivation steps to derive our example trees.

If, on the other hand, we want to have the simplest derivation possible, we could simply encode the two complete trees above as elementary trees directly. In this approach every derivation takes only a single step, producing the entire tree by expanding the root; however, such rules are not at all generalizable: they only work for exactly these sentences with these syntactic structures rather than consisting of reusable components.

In between these two extremes are a variety of elementary trees which capture the trade-off between rule reusability and derivation complexity. For example, the second set of rules presented in Figure 7 represents a good level of abstraction, showing how the verbs relate to their subject and object in these *lexicalized elementary trees*.

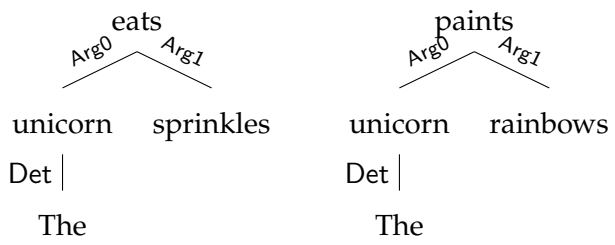
lexicalized elementary tree: *an elementary tree containing one or more words*

In order to derive the full trees for our example, we begin with one of the elementary trees rooted at S, which gives us the verb with its argument structure. At each of the NP leaf nodes, then, we *substitute* one of the elementary trees rooted at NP.

substitute

Of course we may be interested in *dependency trees* where the edges are labelled as well and each node is in fact a word. Consider the following dependency tree versions of the sentences above.

dependency trees



For the *phrase structure* trees shown above, we required the roots of the elementary trees to match the leaf nodes they were replacing. In our elementary trees for the dependency grammar, on the other hand, we will require an *unlabelled* node at the substitution site, and rely on a separate annotation to say where each tree can be substituted.

phrase structure: *here, the structure of a sentence in terms of constituent components, or phrases, which can be identified through substitution tests*

In these trees we represent frontier nodes (also known as *substitution sites*) as empty, unlabelled nodes. We do not, however, indicate any restrictions on where the elementary trees can be substituted, as we can choose different kinds of restrictions for different cases. In this case for example, *The unicorn* makes a much better Arg0 (i. e. agent) than either *sprinkles* or *rainbows*, so one possibility would be restricting where these elementary trees can be substituted based on the label of the arc leading to their substitution site.

substitution sites

With this introduction in place, we can now shift our focus to a formal definition of TSGs.

3.1.1 *Formal Definitions*

For our formal definition of TSGs we build on the notation developed in Eisner (2003), with some adaptations.

Let's begin just by defining a *tree*. A tree t is a tuple of vertices and edges $\langle V, E \rangle$, where V is the set of vertices (also known as *nodes*) and $E \subset V \times V$ is the set of (directed) edges between nodes in V, such that:

tree

nodes

- no node has more than one incoming edge, i. e. $\forall v \in V$ there is at most one vertex $u \in V$ such that $(u, v) \in E$;
- there is exactly one node, called the *root* $r \in V$, which has no incoming edge, i.e. $\nexists u \in V$ such that $(u, r) \in E$; and
- the set of vertices is fully connected by the set of edges, i.e. $\forall v \in V \exists u \in V$ such that $(u, v) \in E$ or $(v, u) \in E$.

root

$$
\begin{aligned}
V &= \{v_0, v_1, v_2, v_3\} && (1)\\
E &= \{(v_0, v_1), (v_1, v_2), (v0, v3)\} && (2)\\
L &= \{\mathsf{Arg0}, \mathsf{Arg1}, \mathsf{Det}, \textit{eats}, \textit{sprinkles}, \textit{unicorn}, \textit{The}\} && (3)\\
l &= \{l(v_0) = \textit{eats}, l(v_1) = \textit{unicorn}, l(v2) = \textit{The}, l(v_3) = \textit{sprinkles},\\
&\qquad l(v_0, v_1) = \mathsf{Arg0}, l(v_1, v_2) = \mathsf{Det}, l(v_0, v_3) = \mathsf{Arg1}\} && (4)
\end{aligned}
$$

Table 3: Formal representation of our first example dependency tree.

labelled tree

Since we are interested in representing linguistic structures with our trees, we would like to associate a *label* with at least some of the nodes in our tree structures. A *labelled tree* is a tree with a set L of possible labels for nodes and edges in the tree and a labelling function $l : V \cup E \rightarrow L$ for associating nodes and edges with particular labels. Hence a labelled tree is a tuple $\langle V, E, L, l \rangle$ (see Table 3).

elementary trees

states

Finally we are prepared to define the *elementary trees* we need for a tree substitution grammar. An elementary tree is a labelled tree which allows its roots and leaves to be associated with *states* Q. These states are annotations indicating how elementary trees can be combined with one another according to the rules of TSGs.

In order to incorporate these states, we define an elementary tree as a tuple $\langle V, V^i, E, L, l, Q, q, s \rangle$, where:

- V, E, L, and l are as defined above;
- internal or interior nodes: $V^i \subseteq V$ is the set of *internal or interior nodes* for this elementary tree;
- states: Q is a set of *states* which can be assigned to frontier nodes or the root node of an elementary tree;
- root state: $q \in Q$ is the *root state*, the state associated with the root node of this elementary tree; and
- state assignment function: $s : V/V^i \rightarrow Q$ is a *state assignment function* mapping frontier nodes to states.

frontier nodes

For convenience we also define notation for the set of *frontier nodes* $V^f = V/V^i$ of an elementary tree. Table 4 shows formal definitions of the elementary trees rooted at *eats* and *unicorn* from above.

We can now define a tree substitution grammar $G = \langle T, Q, I \rangle$, where:

- $T = \{\text{elementary trees } t_i | t_i.Q \subseteq Q\}$ is a set of elementary trees t_i whose states $t_i.Q$ are a subset of the grammar's states Q;
- Q is a set of states used to guide the derivation process; and
- initial states: $I \subseteq Q$ is a set of *initial states* where derivation can begin.

$$V = \{v_0, v_1, v_2\} \tag{5}$$
$$V^i = \{v_0\} \tag{6}$$
$$E = \{(v_0, v_1), (v_0, v_2)\} \tag{7}$$
$$L = \{\mathsf{Arg0}, \mathsf{Arg1}, \mathit{eats}\} \tag{8}$$
$$l = \{l(v_0) = \mathit{eats}, l(v_0, v_1) = \mathsf{Arg0}, l(v_0, v_2) = \mathsf{Arg1}\} \tag{9}$$
$$Q = \{\text{root}, (\mathit{eats}, \mathsf{Arg0}), (\mathit{eats}, \mathsf{Arg1})\} \tag{10}$$
$$q = \text{root} \tag{11}$$
$$s = \{s(v_1) = (\mathit{eats}, \mathsf{Arg0}), s(v_2) = (\mathit{eats}, \mathsf{Arg1})\} \tag{12}$$
$$---$$
$$V = \{v_0, v_1\} = V^i \tag{13}$$
$$E = \{(v_0, v_1)\} \tag{14}$$
$$L = \{\mathsf{Det}, \mathit{The}, \mathit{unicorn}\} \tag{15}$$
$$l = \{l(v_0) = \mathit{unicorn}, l(v_1) = \mathit{The}, l(v_0, v_1) = \mathsf{Det}\} \tag{16}$$
$$Q = \{(\mathit{eats}, \mathsf{Arg0})\} \tag{17}$$
$$q = (\mathit{eats}, \mathsf{Arg0}) \tag{18}$$
$$s = \{\} \tag{19}$$

Table 4: Formal descriptions for two elementary trees. The first for the tree rooted at *eats* with two frontier nodes at the end of arcs labelled Arg0 and Arg1. The second tree has no frontier nodes (as evidenced by the empty function s and the fact that $V = V^i$). This second tree represents the phrase *The unicorn* and has a root state which is only compatible with the first frontier node (v_1) in the first elementary tree. This also means it is incompatible with the elementary tree rooted at *paints* presented in Sec. 3.1.

derivation

We can now use the tree substitution grammar G to derive labelled trees. Derivation begins by selecting an elementary tree $e \in T$ whose root state $e.q$ is in the set of initial states I. For each frontier node $v_j^f \in e$, we then choose an elementary tree $e_j \in T$ whose root state is the same as the state of that frontier node (i.e. $e_j.q = v_j^f.q$) and *expand* that node by *substituting* the tree e_j into e at that node.

expansion and substitution

This substitution operation transforms e into a new tree by:

- replacing $e.V$ with $e.V \cup e_j.V / \{v_j^f\}$;
- replacing $e.V^i$ with $e.V^i \cup e_j.V^i$;
- replacing $e.E$ with $e.E' \cup e_j.E$, where $e.E'$ is $e.E$ with all instances of v_j^f replaced with the root of e_j;
- replacing $e.L$ with $e.L \cup e_j.L$ and $e.l$ with $e.l \cup e_j.l$;
- replacing $e.Q$ with $e.Q \cup e_j.Q$;
- preserving the root state $e.q$ of e; and
- replacing $e.s$ with $e.s \cup e_j.s / \{v_j^f, e_j.q\}$.

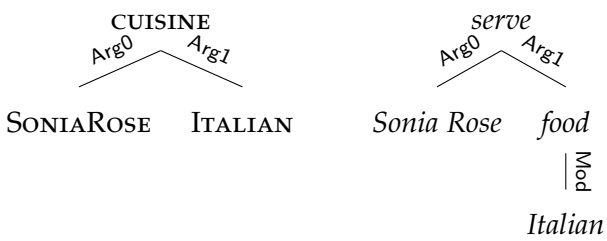
derived tree

A complete *derived tree* is a tree derived by the repeated application of this substitution operation at the remaining frontier nodes of the tree until there are no more frontier nodes left. This formal grounding provides the notational foundation we will build upon throughout the rest of this thesis.

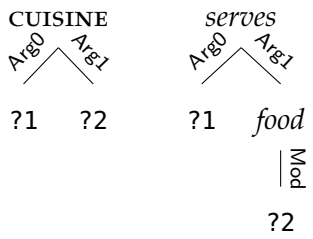
3.2 SYNCHRONOUS TREE SUBSTITUTION GRAMMAR

Now that we understand tree substitution grammars, we can begin to explore *synchronous* tree substitution grammars. A synchronous TSG provides a natural way of connecting two kinds of trees to each other through a joint derivation (e. g., semantic trees and syntactic trees, syntactic trees of one language and those of another language).

Let's begin with an example pair of trees (a *TreePair*) representing the input and the output to a sentence planner.

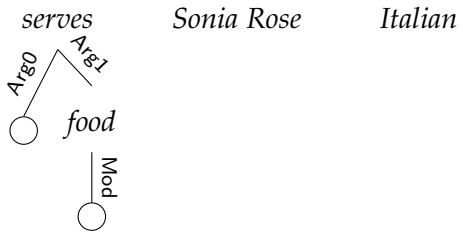


And let's suppose that we have two very small TSGs, one for each of these trees. We'll leave out state annotations for the time being and



list trees in the order in which they combine (i. e. beginning with the leftmost tree, we fill frontier nodes from left to right with subsequent trees). Here we might have the following elementary tree sequences on each side:

and

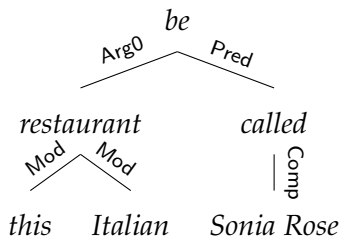


In deriving the complete trees using these elementary trees, the first elementary tree in each derivation has two arguments. In order to make this a joint derivation, we have to pair this elementary tree with an alignment between its frontier nodes and the frontier nodes of the other tree. For this exposition, we will index the frontier nodes with numbers preceded by question marks.

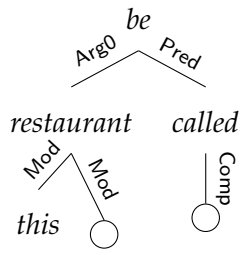
Doing this yields the following as a first derivation step for the joint derivation of the TreePair:

In this case the alignment is trivial: the first frontier node on the semantic side (rooted at CUISINE) is aligned to the first frontier node on the syntactic side (rooted at *serves*), and likewise for the second frontier node in each tree.

Consider, in contrast, expressing this same meaning with the sentence *This Italian restaurant is called Sonia Rose*, which could be represented by the following tree:

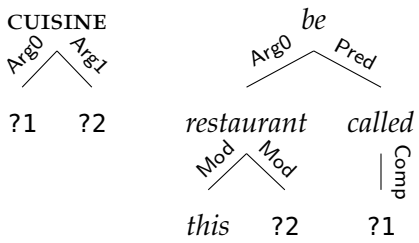


A good elementary tree for deriving the root of this tree would be:



but in pairing this with the input tree from above, we need to reverse our alignments: the first frontier node of the input tree must be aligned to the *second* frontier node of the output tree, and likewise for the second frontier node of the input tree to the first frontier node of the output tree.

Hence the rule:



In this case we can again use the paired elementary trees ITALIAN – *Italian* and SONIAROSE – *Sonia Rose* to complete the derivation.

This example serves to illustrate the most important extension when moving from TSGs to their synchronous counterparts: when pairing the elementary trees of two TSGs, we must specify the alignments between their frontier nodes.

3.2.1 *Formal Definitions*

With this intuition in place, we can now extend our formal definitions from Section 3.1.1 to the synchronous case, again adapting the notation of Eisner (2003).

elementary tree pair

Let's begin by defining an *elementary tree pair* as a tuple $\langle t_1, t_2, q, m, s \rangle$, where:

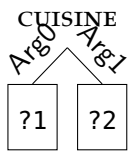
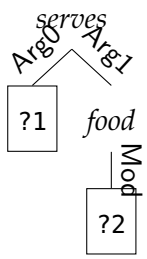
t_1	
t_2	
q	$(\text{root}_{\text{sem}}, \text{root}_{\text{syn}})$
m	$\{m(u_1) = v_1, m(u_2) = v_3\}$
s	$\{s(u_1, v_1) = ((\textsc{cuisine}, \mathsf{Arg0}), (\textit{serves}, \mathsf{Arg0})),$
	$s(u_2, v_3) = ((\textsc{cuisine}, \mathsf{Arg1}), (\textit{food}, \mathsf{Mod}))\}$

Table 5: An elementary tree pair for the first rule presented in this section, representing the trees visually rather than formally. In this case both trees have a state specific to their TSG grammar called 'root'. The nodes labelled ?1 are called u_1 in t_1 and v_1 in t_2, while the node labelled ?2 in t_1 is u_2 and the node with that label in t_2 is v_3.

- $t_j = \langle V_j, V_j^i, E_j, L_j, l_j, Q_j, q_j, s_j \rangle$ represents an elementary tree in one of the component TSGs;
- $q = (q_1, q_2)$ is the pair of root states of t_1 and t_2;
- $m : V_1^f \to V_2^f$ is a bijection pairing each frontier node of t_1 with exactly one frontier node of t_2; and
- $s : V_1^f \times V_2^f \to Q_1 \times Q_2$ such that $s((v_1^f, v_2^f)) = (s_1(v_1^f), s_2(v_2^f))$ maps each matching $m = (v_1^f, v^f 2) \in V_1^f \times V_2^f$ to the pair of states corresponding to the two frontier nodes v_1^f and v_2^f (i. e. $(s_1(v_1^f), s_2(v_2^f))$).

Now we can define a synchronous TSG G as a collection of elementary tree pairs coupled with a set of states and a set of initial states for derivation: $G = \langle \textit{TreePairs}, Q, I \rangle$, where:

- $\textit{TreePairs} = \{\text{elementary tree pairs } \textit{pair} \mid \textit{pair}.s \subseteq Q\}$ is a set of elementary tree pairs whose states s are a subset of the grammar's states Q;
- Q is a set of states used to guide the derivation process; and
- $I \subseteq Q$ is a set of initial states where derivations can begin.

Similar to the process of deriving a tree using a TSG, deriving a tree pair using a synchronous TSG begins by choosing an elementary tree pair whose root state q is in the set of initial states I. Instead of iterating over the individual frontier nodes of each tree, we now iterate over the *matchings*, or alignments, m between the elementary trees. Each matching has a state assigned by s, which we use to select the next elementary tree pair. We continue the process of substituting in elementary tree pairs for each matching between frontier nodes until there are no matchings left.

matchings

The substitution process itself works the same as in the TSG case, simply drawing the trees according to the set of elementary tree pairs and their matchings rather than based on the set of elementary trees for a single TSG and the state of each frontier node. Where the TSG derivation process produced only a single tree, the sTSG derivation process derives two trees simultaneously.

3.3 SUITABILITY OF STSGS FOR SENTENCE PLANNING

With an understanding of the synchronous TSG formalism in hand, we can now explore how well they can capture different kinds of sentence planning rules. This exploration is of course dependent on the kinds of trees which are present in the input and the output representations for our sentence planner.

As a simple linguistic example, the treatment of coordination varies across different approaches to dependency representations, such that the trees present in Figure 8 for the phrase 'good decor and good service' are all valid in different analyses. Such choices can make it more or less challenging to specify a semantically coherent derivation for the tree representing a given utterance as part of a synchronous derivation.

In the sections that follow, we will focus on the strengths and weaknesses of sTSGs for the actual input and output representations used to develop our system (cf. Sections 6.2.1 & 6.2.2).Along the way we will try to highlight the generalizations that follow from our particular observations, so that it is possible to see where alternative input or output representations may be stronger or weaker than those we are using.

3.3.1 *Lexicalization*

Let's begin by exploring the most basic function of the sentence planner: choosing what words should be used to express the meanings in the input text plan. The example from Sec. 3.2 (repeated here for convenience) provides a starting point. For this tree:

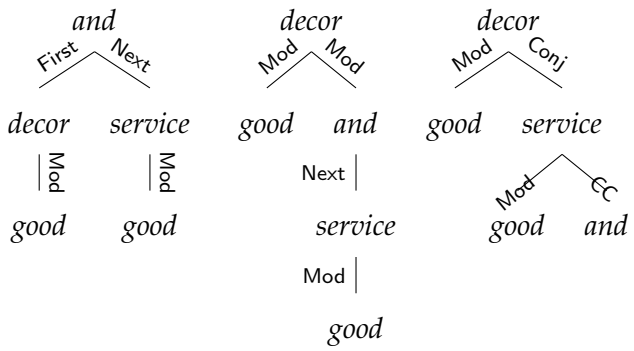
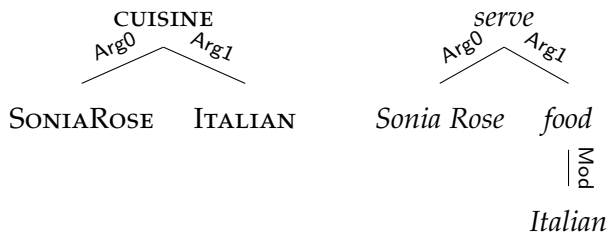
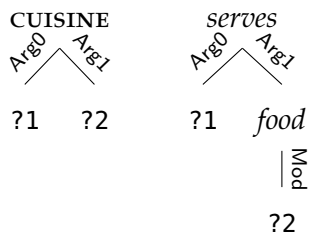


Figure 8: Three approaches to simple coordination of two adjective+noun noun phrases (NPs). The first resembles the approach taken in OpenCCG's grammar based on CCGbank (both introduced in Sec. 6.2.1). The second treats the first NP as the head and uses the conjunction *and* as an intermediate node bridging the two NPs. The third resembles the treatment taken in the Universal Dependencies annotation scheme (de Marneffe et al., 2021).



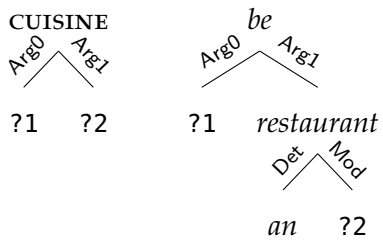
we can have the following rules:



SoniaRose *Sonia Rose* Italian *Italian*

which allow us to express the CUISINE relation with the verb *serves* and to realize the entity name and the adjective describing the cuisine in the natural way. However, we could also realize the same meaning by saying *Sonia Rose is an Italian restaurant*, as in:

Now for this tree we need a new rule for CUISINE, relating it to a sentence rooted at *be*:



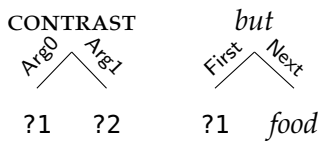
but we can re-use the rules created for SoniaRose and Italian.

These examples show that lexicalization can be quite easily specified in an sTSG. However, determining where to split a pair of trees into rules is not always so straightforward. As we shall see in exploring issues related to aggregation, moreover, the sTSG formalism often forces us to lexicalize more of the structure than we would like, leading to less generalizable rules.

3.3.2 *Aggregation*

Figure 9 highlights two cases of aggregation encoded at the input level by the CONTRAST relation.

In the first tree it is clear how to write an appropriate sTSG derivation, analogously to the rules in for CUISINE in the previous section:



However, this is not so straightforward for the second tree (bottom). In particular, the roots of the two trees must be aligned in an sTSG, requiring us to align CONTRAST to *be*. While we can write several sTSG rules compatible with this tree pair, they all require the grouping of at least CONTRAST $\xrightarrow{\text{Arg1}}$ PRICE, as in:

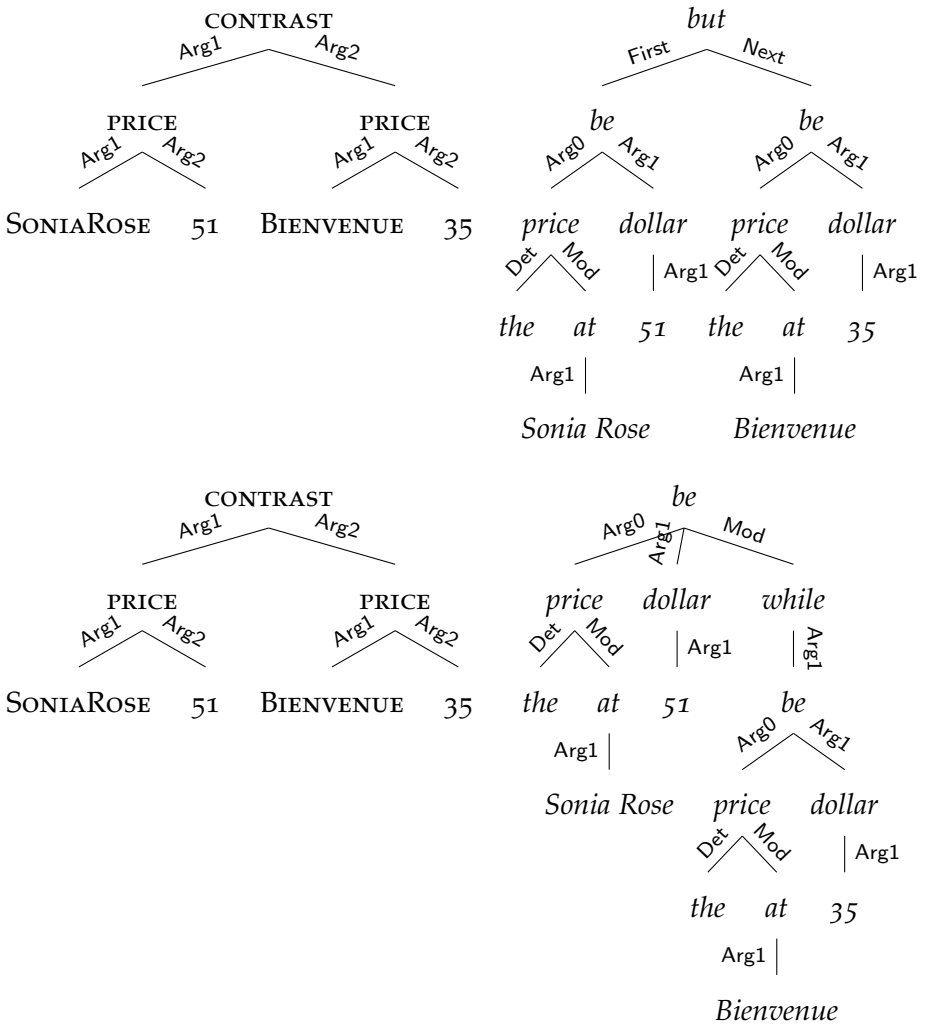
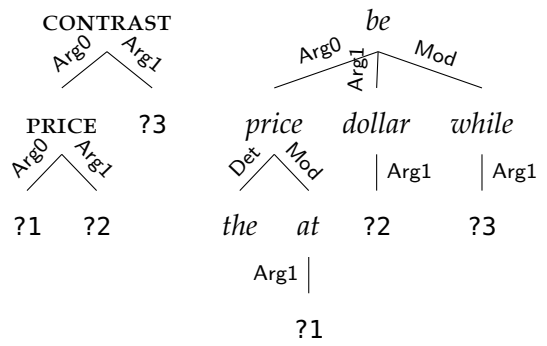


Figure 9: Realizations of the CONTRAST relation with the words *but* (top) and *while* (bottom).



Such trees clearly miss a useful generalization, as we cannot represent the idea that *while* expresses contrast in other contexts and we will need a separate rule to express CONTRAST using *while* when the first argument of CONTRAST is something other than PRICE (e. g. CUISINE, DECOR, etc.).

There are other challenging cases for aggregation rules, such as specifying rules for coordination at the level of verb phrases or conjoining arguments to a shared verb (e. g. turning 'The restaurant has good decor. The restaurant has very good service' into 'The restaurant has good decor and very good service'). However, these challenges have the same outcome: a rule must be overspecified (and therefore less generalizable), as above; or a rule can be underspecified (and therefore end up applying in places it should not, where it produces a semantic or syntactic anomaly).

3.3.3 *Referring Expression Generation*

Finally we can comment on the adequacy of sTSGs with respect to referring expression generation. While it is possible to use pronominalization by, in our running examples, having a rule pairing SONIAROSE with *it*, *this one*, or *that one*, these rules cannot encode the requirement that pronominals require an antecedent to be licit.[1] The only way to enforce such a constraint in this sTSG formalism would be to have rules representing much larger subgraphs. For example, we could create an overly specific rule as in Figure 10.

In Section 3.2, we introduced a rule which included an alternative referring expression to express the kind of cuisine served at a restaurant, 'this *Adjective* restaurant'. In that rule, the referring expression is simply a part of the larger elementary tree used to express this fact. However, it is also possible to create ansTSG rule which allows the use of such expressions as the subject of a sentence expressing another proposition, as in Figure 11.

We have intentionally written this rule with ?1 and ?3 indexed separately, due to the aforementioned lack of a way to restrict rules to

1 Ignoring, of course, instances of cataphora.

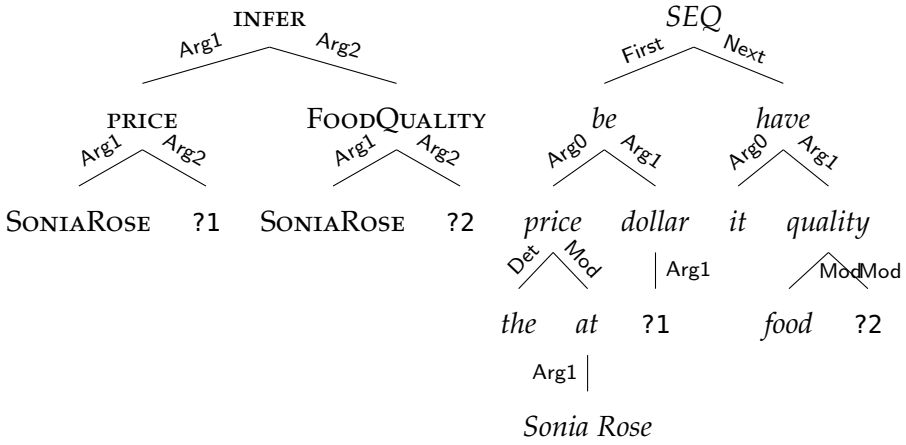


Figure 10: An sTSG rule for pronominalization in a very particular context. INFER is an underspecified additive discourse relation between its arguments, and *SEQ* represents a sequence of sentences.

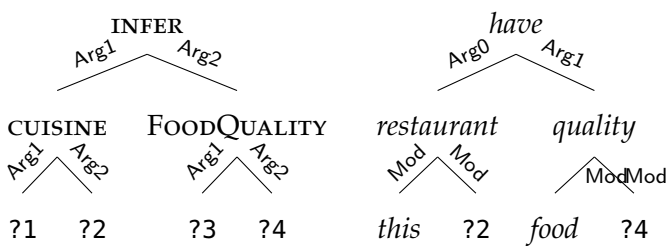


Figure 11: An sTSG rule allowing the use of 'this ?2 restaurant' as the subject of sentences making assertions about food quality.

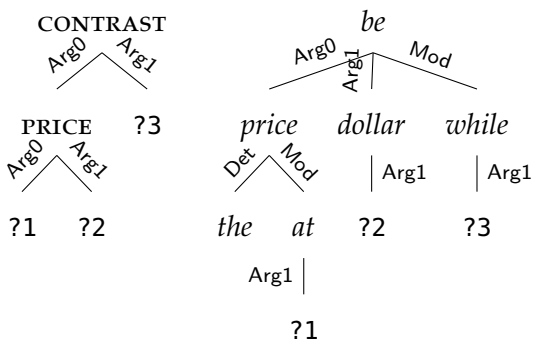
require the same arguments in multiple positions. This rule, therefore, is both oddly underspecified (allowing the CUISINE and FOODQUALITY to have different subjects) and oddly overspecified (restricting the second proposition to be FOODQUALITY and be expressed by the 'has ?4 food quality' construction).

Feature-based TSGs and Tree Adjoining Grammars (TAGs) may be able to encode such restrictions, allowing us to use a feature to enforce coreference constraints. However, inducing sTSGs already presents substantial challenges, as we will see in Chapter 9, so we leave such exploration to future work.

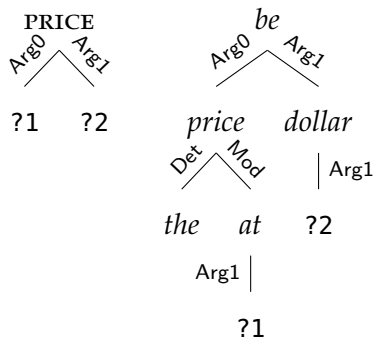
3.4 INTRODUCING (SYNCHRONOUS) DEPENDENCY ATTACHMENT GRAMMARS

So far we have concerned ourselves with tree grammars which only allow substitution at terminal nodes to derive trees. While we have seen that a number of sentence planning tasks can be expressed using the synchronous variant of such grammars, we have also found a number of cases where limiting ourselves to substitution will either result in a proliferation of similar elementary tree pairs to express slightly different semantics or in under-constrained rules.

Consider again the elementary tree pairs for realizing CONTRAST with *while* in Section 3.3.2:

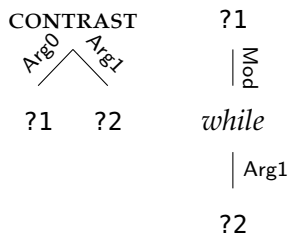


However, we also need a rule like the following to realize PRICE in other contexts:

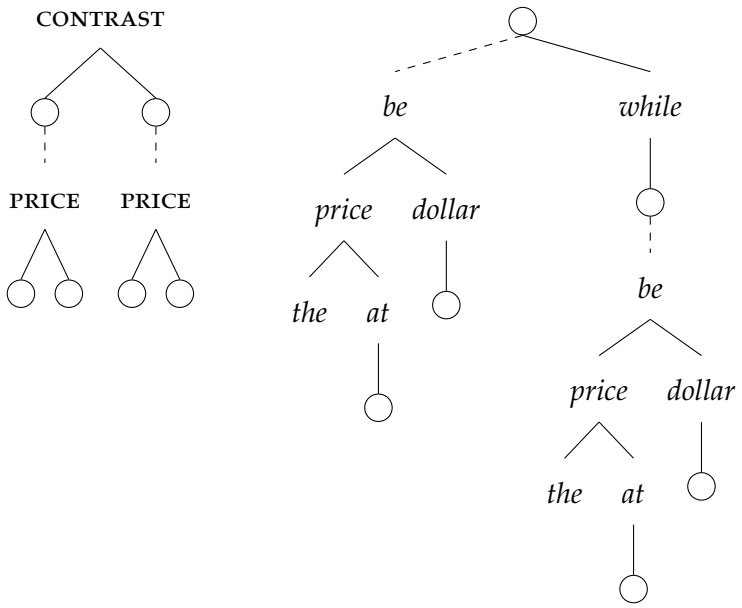


So it would be convenient if we had a way to associate CONTRAST and *while*, other than just creating a version of the first rule for every other proposition (e. g. CUISINE, DECOR)..

Dependency Attachment Grammar (DAG)[2] give us just this ability. Instead of requiring that empty nodes appear only at the frontier of an elementary tree, a DAG allows us to have *internal* nodes which are unlabelled and therefore potential substitution sites. This allows us to have a rule like:



Now our derivation can begin with this elementary tree pair, inserting two of our ideal tree pairs for PRICE as shown below.[3] In this derivation we begin with the treepair we just specified, and so the root of the right-hand tree is initially unlabelled. The alignment tells us where to attach the trees rooted at *be* based on where the PRICE nodes attach in the left-hand tree.



The ability to have unlabelled nodes appear in non-leaf positions makes it easier to write a compact grammar expressing the kinds of operations we need in lexicalization and aggregation.

2 'Dependency Attachment Grammar' is our name for the version of the formalism sketched in Joshi & Rambow (2003) used in this thesis.

3 Here we will omit the indices for arguments and instead indicate where different elementary trees attach with dashed lines. We also omit arc labels for readability.

3.4.1 *Formal Definition*

We will adapt the formal definition given for TSGs above, renaming 'frontier nodes' to 'attachment sites' and explicitly allowing non-terminal nodes to serve as attachment sites.

We define a dependency attachment grammar elementary tree as a tuple $\langle V, V^l, E, L, l, Q, q, s \rangle$, where:

- V, E, L, and l are vertices, edges, possible labels, and a labelling function as defined for TSGs;

labelled nodes

- $V^l \subseteq V$ is the set of *labelled nodes* for this elementary tree;

states

- Q is a set of *states* which can be assigned to unlabelled nodes or the root node of an elementary tree;

root state

- $q \in Q$ is the *root state*, the state associated with the root node of the elementary tree; and

state assignment function

- $s : V/V^l \rightarrow Q$ is a *state assignment function* mapping unlabelled nodes to states.

With this definition of elementary trees, we can use the same definition that we used for TSGs earlier, defining a dependency attachment grammar $G = \langle T, Q, I \rangle$, where:

- $T = \{\text{elementary trees } t_i | t_i.Q \subseteq Q\}$ is a set of elementary trees whose states $t_i.Q$ are a subset of the grammar's states Q;
- Q is a set of states used to guide the derivation process; and

initial states

- $I \subseteq Q$ is a set of *initial states* where derivation can begin.

attachment *derivation*

The superscript f *now stands for 'free', meaning unlabelled.*

In place of the substitution operation used for TSGs, we use the more general *attachment* operation. Derivation begins by selecting an elementary tree $e \in T$ whose root state $e.q$ is in the set of initial states I. For each unlabelled node $v_j^f \in e$, we then choose an elementary tree $e_j \in T$ whose root state is the same as the state of that unlabelled node (i.e. $e_j.q = v_j^f.q$) and *attach* the new elementary tree e_j to e at that node. Attachment means identifying the two nodes with each other, preserving all parents and children.

This attachment operation transforms e into a new tree by:

- replacing $e.V$ with $e.V \cup e_j.V/\{v_j^f\}$;
- replacing $e.V^l$ with $e.V^l \cup e_j.V^l$;
- replacing $e.E$ with $e.E' \cup e_j.E$, where $e.E'$ is $e.E$ with all instances of v_j^f replaced with the root of e_j;
- replacing $e.L$ with $e.L \cup e_j.L$ and $e.l$ with $e.l \cup e_j.l$;

- replacing $e.Q$ with $e.Q \cup e_j.Q$;
- preserving the root state $e.q$ of e; and
- replacing $e.s$ with $e.s \cup e_j.s/\{v_j^f, e_j.q\}$.

Note that, while the final two parts of the operation may appear to be in conflict when the root of e is an unlabelled node, the state of this unlabelled node is the same as the labelled node being attached to it. Therefore removing the unlabelled node v_j^f from the domain of the state assignment function does not remove the state associated with the root, since the root of the derived tree is the *labelled* node added from e_j.

A complete *derived tree* is a tree derived by the repeated application of this attachment operation at the remaining unlabelled nodes until there are no more unlabelled nodes. derived tree

Now we can update our definition of *elementary tree pairs* as a tuples of the form $\langle t_1, t_2, q, m, s \rangle$, where: elementary tree pairs

- $t_j = \langle V, V_j^l, E_j, L_j, l_j, Q_j, q_j, s_j \rangle$ represents an elementary tree in one of the component dependency attachment grammars;
- $q = (q_1, q_2)$;
- $m : V_1^f \rightarrow V_2^f$ is a bijection pairing ensuring that every unlabelled node in t_1 with is aligned with exactly one node in t_2 and that every unlabelled node in t_2 is aligned with exactly one node in t_1; and
- $s : V_1^f \times V_2^f \rightarrow Q_1 \times Q_2$ such that $s((v_1^f, v_2^f)) = (s_1(v_1^f), s_2(v_2^f))$.

And our definition of a synchronous dependency attachment grammar follows our established pattern, $G = \langle TP, Q, I \rangle$, where:

- $TP = \{\text{elementary tree pairs } (t_1, t_2) | s \subseteq Q\}$ is a set of elementary tree pairs whose states s are a subset of the grammar's states Q;
- Q is a set of states used to guide the derivation process; and
- $I \subseteq Q$ is a set of initial states where derivations can begin.

As with synchronous TSG derivation, we begin by choosing an elementary tree pair whose root state q is in the set of initial states I and iterate over the matchings between the elementary trees. In this case, our matchings are designed to allow for *attachment* in addition to substitution. That is, a *labelled* node may be matched to an unlabelled node, in which case derivation can only proceed using a rule with an unlabelled root node which can attach to the labelled node. attachment

3.5 CONCLUSION

In this chapter we have seen how two related grammatical formalisms might be used to represent sentence planning rules: synchronous tree substitution grammars and synchronous dependency attachment grammars. While our approach to dependency attachment grammars is similar to the one sketched in Joshi & Rambow (2003), this thesis contributes a formal definition for the original formalism along with a novel synchronous variant.

Now that we see how the kinds of sentence planning rules we are interested in can be captured by a synchronous grammar, we can begin thinking about how to automatically learn such grammars. To do that, we will need to first understand some statistical machinery, which we turn to next.

4 ESTIMATING PROBABILITY AND THE CHINESE RESTAURANT PROCESS

This thesis takes an approach to machine learning which is rooted in probability models, so we need to understand a bit about theoretical and empirical models of probability, as well as ways of combining these models. This chapter introduces these concepts at a high level before diving into more detail for methods used in this thesis: namely, the Chinese Restaurant Process and Gibbs sampling. We conclude with a discussion of the benefits and drawbacks of these methods for modelling linguistic phenomena.

With this background we are prepared for the next chapter, which addresses how these and related methods can be used for grammar induction. Our own theoretical framework (Ch. 6) and the models explored in this thesis (Ch. 9) will also build on this understanding.

4.1 THEORETICAL AND EMPIRICAL PROBABILITY ESTIMATES

Let's begin with an all-too-classic example: we want to reason about the possible outcomes when flipping a coin. To begin with, we'll ignore the possibility that the coin could end up balanced on its edge and consider only two possible outcomes, which we will call HEADS and TAILS, referring to the obverse or the reverse of the coin being visible when the coin lands, respectively.

What is the probability that flipping a coin will result in a particular outcome? In this case we know that we can only have one outcome occur out of two possible outcomes (for a single coin flip), so we can say based on the space of possible outcomes that the probability is $\frac{1}{2} = 50\%$.

But this assumes that both outcomes are equally likely! Why would we assume this? One good reason, we could say, is that most of the coins we have seen in our lifetimes are *fair* coins: that is, they are not biased toward one outcome or the other. But most coins have imperfections, especially those that have been in circulation for a long time, so maybe there is some small bias in one direction or the other. Or maybe we are dealing with a nefarious purveyor of *biased* coins which always come up HEADS.

To deal with this uncertainty, we could give up on our *theoretical* model and try an *empirical* approach to estimating this probability. For example, we could flip the coin ten, twenty, a hundred, a thousand, or more times and count the number of times the coin comes up HEADS

random variable

In general, for the probability P of a random variable X taking on a particular value x we have:

$$P(X = x) = \frac{1}{\text{number of possible outcomes}} \tag{20}$$

when each outcome is equally likely. The so-called random variable represents a set of possible outcomes and the notation $X = x$ means that the random variable has taken on the particular value x from the set of possible outcomes. This set of possible outcomes is called the support of the probability distribution P over the random variable.

support

versus the number of times the coin comes up TAILS. In this case our estimate of the probability is:

$$P(\text{coin} = \textsc{heads}) = \frac{\text{number of times we observed } \textsc{heads}}{\text{number of observations we made}} \tag{21}$$

theoretical
empirical

So we have two ways of estimating the probability of a coin coming up HEADS: a purely *theoretical* approach and a purely *empirical* approach. The theoretical approach clearly breaks down if any of our assumptions about the fairness of the coin are broken, but getting an accurate estimate of this probability in an empirical approach could require a large number of observations, so it would be nice if we had a principled way to combine these approaches.

4.2 INTERPOLATING PROBABILITY MODELS

How can we combine these two different approaches to modelling probabilities? One simple option is to take the mean of the two estimates:

$$P(X = x) = \frac{P_{theoretical}(X = x) + P_{empirical}(X = x)}{2} \tag{22}$$

This places equal weight on the two estimates of probability; however, we may want to give more weight to the theoretical estimate or to the empirical estimate. Fortunately, we can rewrite Equation 22 in a more general form as

$$P(X = x) = \frac{a}{a+b} P_{theoretical}(X = x) + \frac{b}{a+b} P_{empirical}(X = x) \tag{23}$$

to give more weight to one estimate of the probability or the other.

4.3 INFINITELY MANY POSSIBLE OUTCOMES

Our example so far has been the probability of a coin coming up heads or tails, but language allows for much more variation than this. In theory, the productive capacity of language is endless, and in practice it is difficult to delimit exactly what linguistic structures a statistical model should cover.

One solution to this problem is to write a *generative model*, which describes how to generate the structures in question and how to calculate the probabability based on this generative process. With such a model we can calculate the probability of any particular structure, even when it is impossible to explicitly enumerate all possible structures the model can generate.

generative model

For a simple linguistic example, let's consider adjectival modifiers in English noun phrases. When deciding how to describe a very sparkly unicorn, the grammar of English allows us to use the word *very*, and similar modifiers, as many times as we would like. So we could say...

- the sparkly unicorn
- the very sparkly unicorn
- the very, very sparkly unicorn
- the very, very, very sparkly unicorn
- the very, very, ... very sparkly unicorn

One simple theoretical probability of this sequence of words is to multiply the probability of all of the words together. Let's say that the probability of each unique word type in this noun phrase is $\frac{1}{4}$. Then for the above phrases we have the probabilities: $\frac{1}{64}$, $\frac{1}{256}$, $\frac{1}{1024}$, and $\frac{1}{4096}$, before we reach the non-enumerated case.

While we cannot write out the probability for every possible outcome, because there are infinitely many of them, we can write an equation for the probability of one of these noun phrases based on the number of times the word *very* appears in it. Specifically, we have:

$$P(\text{the very, ... very sparkly unicorn}) = \frac{1}{64}(\frac{1}{4})^n \tag{24}$$

where n is the number of times the word *very* occurs in the phrase.

This model implicitly assumes that the choice of each word is independent of the words preceding and following it and the meaning of the text[1], so it is not a very good model of human language[2], but it serves to illustrate the point: generative models can allow us to define probability distributions over infinitely many possible outcomes.

1 And that a human would have the patience for more than two or three *very*s...
2 or even English language

4.4 THE CHINESE RESTAURANT PROCESS

A Chinese Restaurant Process (CRP) allows us to build upon the observations of the previous two sections, providing a way of beginning with a theoretical, generative model of some phenomenon and interpolating between this model and empirical observations.

Intuitively, before we have any observations we can only base our expectations on our theoretical model. We call this model the *base distribution* of the CRP we are defining.[3] As we make more observations, however, we would like to place more weight on our empirical estimate of the probability. Let's make this explicit in rewriting Equation 23 with $a = \alpha$ (the *concentration parameter*, as explained in the next subsection) and $b = N$, the number of observations we have made so far:

base distribution

$$P(X = x) = \frac{\alpha}{\alpha + N} P_t(X = x) + \frac{N}{\alpha + N} P_e(X = x) \tag{25}$$

Since our empirical estimate of the probability of $X = x$ is simply the number of times we have observed that outcome ($\text{freq}(X = x)$) divided by the number of observations we have made (N), we have:

$$P(X = x) = \frac{\alpha}{\alpha + N} P_t(X = x) + \frac{\text{freq}(X = x)}{\alpha + N} \tag{26}$$

From this equation we can see that our intuition is satisfied: when $N = 0$, $P(X = x) = P_t(X = x)$. Moreover, so long as α is constant, our new probability estimate will converge to the actual empirical estimate as our number of samples approaches infinity.

4.4.1 *Tables in the Chinese Restaurant Process*

So far we have shown how the CRP interpolates between a theoretical base distribution and emprical observations and discussed how we can use this to refine a theoretical model over infinitely many possible outcomes, but now we need to understand... what does this have to do with Chinese restaurants?

When entering a Chinese restaurant in some parts of the world, it seems that there are an endless number of tables to choose from. When someone enters the restaurant, they proceed to the first table with 100% probability. Each subsequent customer then faces a choice: will they start a new table or will they join one of the existing tables? We define the probability of sitting at a new table as $\frac{\alpha}{\alpha+N}$, so the probability of sitting at an existing table is $\frac{N}{\alpha+N}$ (since we have only two options). This is where we see why α is called the *concentration parameter*: when α is larger, arriving customers are more likely to begin a new table, while smaller values of α lead arriving customers to be

concentration parameter

3 If you are talking to a Bayesian, you might also hear them call it their *prior*.

more concentrated, preferring to sit at the tables which already have customers.

If a customer chooses to start a new table, they take a seat and choose a dish from the menu with probability $P_t(\text{DISH} = \text{dish}_i)$. If the customer chooses to sit at one of the existing tables, they will sample a decision according to the popularity of the table.That is, the probability a customer who is choosing an existing table choosing a particular table is the number of customers sitting at that table divided by the total number of customers in the restaurant. In our example, each of the tables is associated with the observation of a particular value x for the random variable X.

This means that our earlier equations were slightly simplified. $P_e(X = x)$ should be the sum over tables where the label for the table $X = x$:

$$P_e(X = x) = \sum_{t \in \{\text{tables} | \text{label } X = x\}} \text{count}(t), \tag{27}$$

where $\text{count}(t)$ is the number of customers seated at table t.

When the base distribution for the Chinese Restaurant Process is not dynamic—that is, when it is a fixed distribution and not itself updated by new observations—we can in fact get away with ignoring the tables altogether and keeping track only of the number of observations with $X = x$. However, when we are using a *hierarchical* CRP, each time we sample a label for a new table (i.e. each time we draw from the base distribution) we are adding an observation to that base distribution.

This becomes important during training when we are adding and removing observations from the upper CRP, because each time we remove an observation we must remove it from the specific table with which it is associated in order to know when that table is empty and therefore when to remove the observation associated with that table from the lower CRP.

4.5 FITTING MODELS WITH GIBBS SAMPLING

Our models will frequently be more complex than the ones used as examples in this chapter. In particular, we often model a large joint distribution using *latent variables* to represent factors which we cannot directly observe. Gibbs sampling provides one way of fitting such models.

latent variables

To make the example concrete, let's think about the challenging task of *word segmentation*: when we learn a language, we need to learn to identify individual, discrete words from continuous streams of sound.[4]

word segmentation

4 This example is based on Goldwater, Griffiths & Johnson (2006) and Goldwater, Griffiths & Johnson (2007).

Given a sequence of sounds /ju·wɑnə·si·ðʌ·bʊk/, our challenge is to identify the word boundaries (indicated here with ·). Suppose we have a base distribution over all possible words in the relevant language $P_b(w)$ which we use in a Chinese Restaurant Process $P(w)$ with $\alpha = 1$. The probability of our (corpus of) sound sequences will be the product of the probability of each of the words in the sequences, but we do not know a priori what the true word segmentation is.

Let's imagine for our sequence of sounds we are trying to decide if the first word boundary is valid or not, so we will mark it with a question mark: /ju?wɑnə·si·ðʌ·bʊk/. Using the Expectation-Maximization (EM) algorithm, we would begin by removing the current observation from the model. (That is, we would remove the word /juwɑnə/ or the words /ju/ and /wɑnə/ from the CRP.) Then in the expectation step we would calculate the probability of /juwɑnə/ occurring as a single word versus the sequence of words /ju·wɑnə/ according to the rest of the model (which has presumably been trained on more than this single example). In the maximization step, then, we would choose whichever segmentation had the highest probability according to our model.

Gibbs sampling is like EM, but it replaces the maximization step with a sampling step. Instead of taking the decision which maximizes the probability of the data according to the current state of our model, we *sample* the decision based on the relative probability of the two outcomes. One of the benefits of this approach is that it reduces overfitting of the model to the training data. Another is that this sampling-based procedure converges to the true posterior distribution based on our prior and the observable variables (i. e. our corpus of sound sequences).

Note that this is no small feat. The state space of all possible segmentations explodes as we increase the size of the dataset, so exploring all of it is intractable. This sampling procedure, however, guides our search toward the right part of this state space, usually in a reasonable number of iterations.

4.6 WHY USE THESE METHODS TO MODEL LANGUAGE?

In review, we have mentioned a few traits of these tools which are beneficial to our modelling tasks.

- *Generative models* can model distributions over potentially infinite support.
- *Gibbs sampling* converges to the 'right' distribution based on our model and resists overfitting.
- The *Chinese Restaurant Process* allows us to interpolate between our prior expectations and observed data.

Moreover, the CRP provides a good way to model the tension between the need for smaller Tree Substitution Grammar (TSG) rules (or shorter words) and the need for shorter derivations (or sentences containing fewer words). Using a prior which is biased toward smaller rules captures one aspect of this balance while the caching process of the CRP (the empirical probability model) allows larger rules (or words) to be kept if they are sufficiently helpful in modelling the data.

The tendency of 'customers' in the CRP to clump together further serves to model the well-known Zipfian tendencies of language with its rich-get-richer behavior.

4.7 CONCLUSIONS

This chapter has presented some of the key tools used throughout this thesis: the Chinese Restaurant Process and Gibbs sampling. The next chapter will expand upon this discussion by discussing previous models for learning (synchronous) Tree-Substitution Grammars (cf. Chapter 3) which were used for parsing, machine translation, and text summarization.

5 BAYESIAN APPROACHES TO GRAMMAR INDUCTION

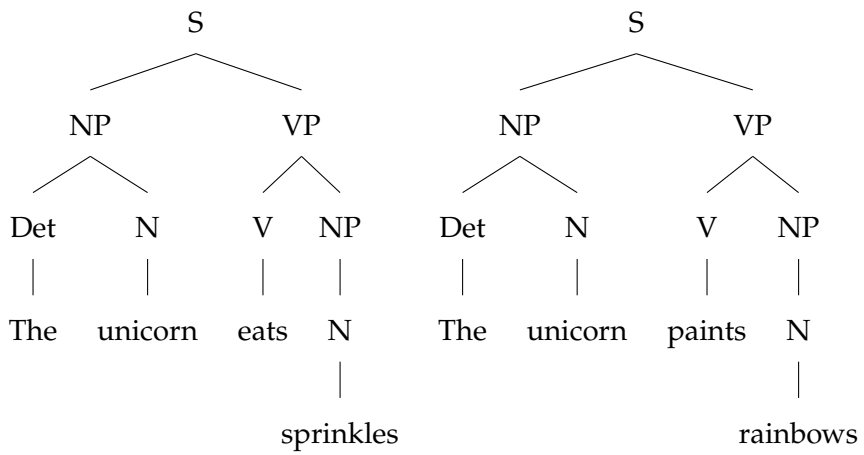
Now that we are familiar with (synchronous) Tree Substitution Grammars (TSGs), the Chinese Restaurant Process (CRP), and Gibbs sampling, we can examine previous work which used these tools for grammar induction.

We begin with the monogrammatical case, where the task is to induce a context-free grammar or a tree-substitution grammar, using the work of Cohn, Blunsom & Goldwater (2010) as an example. Then we shift to the synchronous case, examining the work of Yamangil & Shieber (2010) on sentence compression. Following each of these sections we also present a brief overview of other related work.

The chapter concludes with a discussion of how the current work differs from these approaches, preparing the reader to understand the models presented in Chapter 9.

5.1 INDUCING TREE SUBSTITUTION GRAMMARS

Recall the first trees we presented in Section 3.1, which were simple phrase structure trees for a pair of simple sentences.



These are the kinds of trees for which Cohn, Blunsom & Goldwater (2010) sought to induce a tree-substitution grammar. Remember that a TSG consists of a collection of elementary trees T, a set of states for guiding derivations Q, and a set of initial states $I \subseteq Q$ where derivations can begin. For phrase structure trees like these the order of child nodes is important, but the arcs themselves are not typically

labelled.[1] In TSGs over phrase structure (PS) trees is also typical to assume that the set of states Q for guiding derivations must be a subset of the set of non-terminal labels in the grammar (i.e. {S, NP, VP, Det, N, V} in this example). It is also generally assumed that $I = \{S\}$, where S is the PS category for a sentence.

5.1.1 *A probabilistic model of TSGs*

Probabilistic TSG

A *Probabilistic TSG* (PTSG) further associates with each elementary tree $e \in T$ for each category $c \in Q$ a probability $P(e|c)$, which is the conditional probability of the tree e substituting into a position with the state labelled c. Treating the n derivation steps $\vec{e} = e_1, \ldots, e_n$ in a given tree t as statistically independent, the probability of a particular derived tree is:

$$P(t) = \sum_{\{\vec{e} \mid \text{tree}(\vec{e})=t\}} \prod_{e \in \vec{e}} P(e|l(\text{root}(e))) \tag{28}$$

where $\text{tree}(\vec{e})$ yields the tree resulting from a particular derivation $\vec{e}$. That is, the probability of a particular tree t is the sum of the probabilities of each possible derivation $\vec{e}$ for that tree, with the probability of each derivation taken to be the product of the probabilities of the elementary trees e appearing in that derivation.

Cohn, Blunsom & Goldwater (2010) describe how a Dirichlet Process (DP) can be used to model the distribution over these elementary trees.[2] As with our sparkly unicorn example (cf. Section 4.3), we begin by describing a generative model for the phenomenon we want to model. In this case, we model the distribution over possible elementary trees using the CRP G:

$$G|\alpha, P_E \sim \text{DP}(\alpha, P_E) \tag{29}$$

$$e_i|G \sim G \tag{30}$$

where G is an infinite distribution over possible elementary trees and each e_i is sampled identically and as though identically distributed (*i.i.d.*) from G.

This definition allows us to sample any elementary tree, but there is no way to guarantee that a sampled elementary tree will fit into a particular derivation. Therefore Cohn, Blunsom & Goldwater (2010) define a collection of DPs, each conditioned on its root state $c \in Q$[3]:

1 Rather than extending our formal definition (cf. 3.1.1) to distinguish ordered trees, where the children of a node occur in a fixed order, from the unordered trees previously introduced, we use the labelling function l to annotate edges with positive integers, effectively using the labelling function to encode the order of children.

2 In fact, they describe their model in terms of a generalization of the DP, the Pitman-Yor Process (Pitman & Yor, 1997). For the present work, however, we only need to understand the model in terms of the DP, which we implement as the CRP described in the previous chapter.

3 Note that they actually go a step further than we describe here, also using different values of α for each root state condition c. They also infer values for α_c based on their

$$G_c|\alpha, P_E \sim \mathrm{DP}(\alpha, P_E(\cdot|c)) \tag{31}$$
$$e_i|c, G_c \sim G_c \tag{32}$$

P_E is what we called in the previous chapter our 'theoretical probability distribution', so we have:

$$P(e_i|c) = \frac{\alpha}{\alpha + N} P_E(e_i|c) + \frac{freq(e_i)}{\alpha + N} \tag{33}$$

where N is the number of samples drawn from this DP.

The next step, then, is to define this base distribution P_E. Cohn, Blunsom & Goldwater (2010) define the base distribution based on a Context-Free Grammar (CFG) with rules $c \rightarrow x_1, x_2, \ldots x_n$ defining possible grammar productions. This results in probability distribution:

$$P_E(e|c) = \prod_{c \rightarrow x_1, x_2, \ldots, x_n \in e} P_R(x_1, x_2, \ldots, x_n|c) \prod_{u \in e.V^f} s_{u.q} \prod_{v \in e.V^i} (1 - s_{v.q}) \tag{34}$$

where P_R is the maximum likelihood estimate over CFG production rules in the training corpus and s_c is the probability of ceasing expansion at a node with label c.

This model for P_E ensures that the model prefers small elementary trees, since the s_c terms make the distribution over tree sizes geometric. Coupled with the 'rich-get-richer' tendencies of the CRP, the model is able to find larger elementary trees if they are frequent enough while generally preferring smaller elementary trees.

5.1.2 *Inducing the grammar*

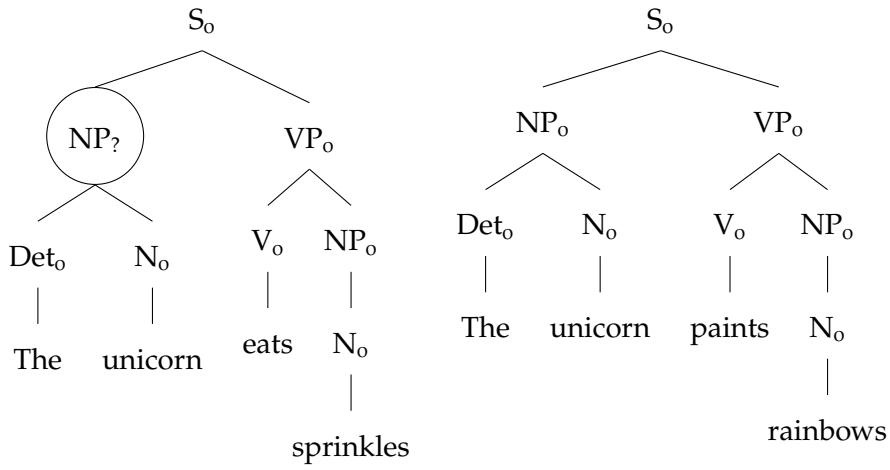
Now that we have a model for the probabilities of different trees in our synchronous TSG, we need to work out how to fit this model based on some training data.

In the last chapter we foreshadowed that we would be using Gibbs sampling for inference, but now we need to define how to update the parameters of a particular model. In working with Phrase Structure Grammar (PSG) trees from the Penn Treebank (PTB), Cohn, Blunsom & Goldwater (2010) observed that the problem of inferring a set of derivations for a given set of trees can be treated as a segmentation problem. This allows them to use a similar approach to Gibbs sampling to the one that Goldwater, Griffiths & Johnson (2006) used for word segmentation (see our example in Section 4.5).

Defining which nodes are substitution sites fully specifies a segmentation of the tree into elementary trees and hence a derivation.

training data using sampling methods we will not use in this thesis and therefore omit from the next section for brevity.

We can associate with each node a binary variable which indicates whether that node is a substitution site or not, which we illustrate using our pair of unicorn trees repeated at the start of Sec. 5.1.



Here we have labelled most of the nodes with 0 to indicate that they are interior nodes for an elementary tree.[4] We have left the first NP unlabelled, however, to ask: how do we weight the two possible values in order to sample appropriately?

This is the decision faced by the *local* Gibbs sampler of Cohn, Blunsom & Goldwater (2010), and we have two hypotheses. Under H_0,we suppose that this node is an internal node. The probability of H_0 based on the current state of this dataset is:

$P($`(S (NP (Det The) (N unicorn)`
`(VP (V eats) (NP (N sprinkles))))))`$)$
$P($`(S (NP (Det The) (N unicorn)`
`(VP (V paints) (NP (N rainbows))))))`$)$

And under the alternative hypothesis H_1 that this is a substitution site, we have probability:

$P($`(S (NP) (VP (V eats) (NP (N sprinkles))))))`$)$
$P($`(NP (Det The) (N unicorn)`$)$
$P($`(S (NP (Det The) (N unicorn)`
`(VP (V paints) (NP (N rainbows))))))`$)$

Since our choice is made by sampling H_0 with probability $\frac{P(H_0)}{P(H_0)+P(H_1)}$ and H_1 with probability $\frac{P(H_1)}{P(H_0)+P(H_1)}$, it is clear that the probability of

4 For simplicity, we treat terminal nodes as fixed: they will always be interior nodes associated with at least one non-terminal node representing their Part of Speech (POS).

the second tree (for the sentence *The unicorn paints rainbows*) will cancel, so we are only interested in the probability of the *merged* elementary trees (H_0) compared to the probability of the *split* elementary tree (H_1). This split-merge decision is at the heart of segmentation-based approaches to Gibbs sampling over linguistic structures.

For this estimate to work, we must remove either the merged trees or the split tree, whichever was previously observed for this decision, before calculating $P(H_0)$ and $P(H_1)$. This prevents a single sampled decision from biasing the model in favor of repeating that decision in future iterations.

After re-sampling this variable, we move on to the next node in the tree and repeat the process, iteratively resampling the split-merge decision for every node in the corpus. One full sweep over the corpus is one *Gibbs iteration*.

Because this approach removes very local observations, resamples a variable, and adds those observations back to the model, it is quite slow and requires a fair amount of bookkeeping. Moreover, getting from one derivation with low probability to another derivation with a higher probability can require going through an extremely low probability series of sampling decisions. By definition these decisions are rare, so the local sampler is said to have limited *mixing*.

mixing*: the extent to which a statistical sampler is able to explore diverse parts of the state space*

Therefore Cohn, Blunsom & Goldwater (2010) propose a *blocked* version of their Gibbs sampler, which remove all observations associated with an entire tree, rather than only those associated with a particular node. We then resample the split-merge decision for each node in that tree, and update our observations only after resampling all of the nodes in the tree.[5]

5.1.3 *Other work on inducing TSGs*

Cohn, Blunsom & Goldwater (2010) showed how their approach could be useful for inducing a TSG given a corpus annotated with PSG trees, but also presented a version which did not require explicit tree structure annotations and could operate on dependency trees.

Cohn, Goldwater & Blunsom (2009), Blunsom et al. (2009), and Cohn & Blunsom (2010) are also related to Cohn, Blunsom & Goldwater (2010), although these authors are not the only ones to look at inducing TSGs. Post & Gildea (2009a,b) also used induced TSGs from the PTB, looking at applications both to parsing and to language modelling.

Shindo, Fujino & Nagata (2011) and Shindo et al. (2012) worked on inducing extended TSG grammars: the former looking at adding an insertion operator and the later working on symbol refinement. Yamangil & Shieber (2013) worked on inducing Tree Adjoining Gram-

5 Hence the name: sampling is done block by block, where each block consists of a single tree.

mars (TAGs) for parsing. Bergen, Gibson & O'Donnell (2015) used the induction of TSGs and TSGs with sister-adjunction added in order to make a learnability argument about the argument-modifier distinction.

These Bayesian approaches were not the first attempts at defining probabilistic TSG models, however. In Data-Oriented Parsing (DOP) (Bod, 1992, 1993; Scha, 1990) the idea is to directly use a parse-annotated corpus as a statistical model for likely elementary trees and perform parsing by sampling from this model. This treats every possible parse of the annotated trees as equally valid and relies on the fact that certain subtrees will appear more often in the corpus to provide a probability estimate for each tree.

Where the Bayesian approaches implicitly model the infinite space of possible elementary trees, the DOP model must model explicitly the exponentially large space of possible parses for a given corpus. For a corpus whose n sentences contain m nodes in their (combined) parse trees, this means the DOP approach must model 2^m possible states. Each sentence has, on average, $2^{\frac{m}{n}}$ possible parses. In DOP all of these are considered equally valid, while in the Bayesian approaches we sample a single state for each sentence (i.e., a single derivation) and therefore only represent n states explicitly at any point in time.

Cohn, Blunsom & Goldwater (2010) summarizes the parameter estimation problems which arise from attempting to use all possible subtrees and highlights some of the strengths of the Bayesian approach. In particular, the explicit generative model with priors biasing the model away from both extremes[6] allows the model to learn a grammar of appropriate complexity for the given training data.

Our approach therefore adopts the Bayesian perspective. We define a generative model for the kind of data we are working with and then use sampling methods to fit that model based on some particular dataset. Understanding Cohn, Blunsom & Goldwater (2010)'s approach to this problem provides the necessary introduction to see how these methods can be extended to the synchronous grammars we need for sentence planning.

5.2 SYNCHRONOUS GRAMMARS

In the synchronous setting our generative story needs to account for pairs of trees as well as the alignments between them. While our work focuses on NLG, our running example in this section will follow related work on sentence compression (Yamangil & Shieber, 2010). The TreePairs for this work are PSG trees similar to the pair pictured in Figure 12, where you can see that some content from the source tree is dropped by this TSG extended with insertion/deletion operations.

6 i.e., reducing to a CFG or deriving each sentence as a single huge elementary tree

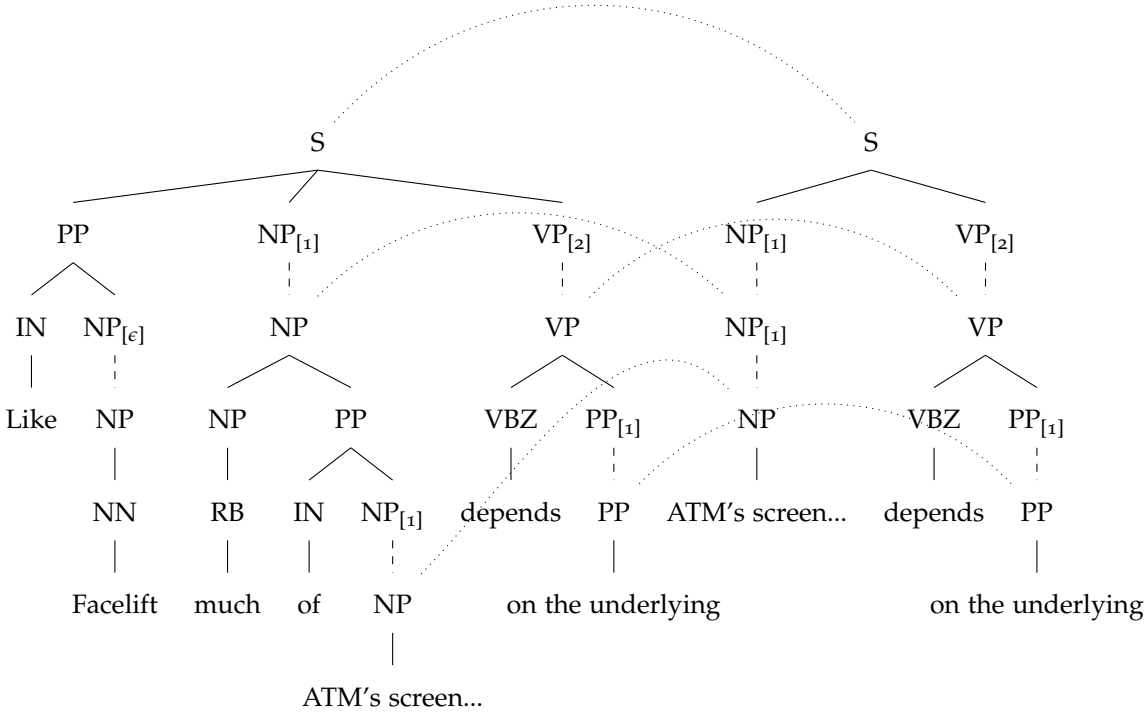


Figure 12: Example TreePair from Yamangil & Shieber (2010). The (un)compressed sentence reads: (*Like FaceLift, much of*) *ATM's screen performance depends on the underlying application*. Note that the repeated $\mathrm{NP}_{[1]}$ node on the right hand side is simply a visualization of the fact that two nodes in the source can be aligned to one node in the target; it is not actually a unary expansion in the underlying grammar.

Their generative model begins with a pair of states q_s, q_t, where s and t stand for *source* and *target* tree, respectively. They model the distribution over possible *pairs* of elementary trees (e_s, e_t) given state q_s, q_t as a CRP:

$$G_{q_s,q_t}|\alpha, P_0 \sim \mathrm{DP}(\alpha, P_0(\cdot|q_s, q_t)) \tag{35}$$

$$(e_s, e_t)|q_s, q_t, G_{q_s,q_t} \sim G_{q_s,q_t} \tag{36}$$

This is basically the same model we described for the synchronous grammar, only using pairs of state labels and sampling pairs of elementary trees. The base distribution, P_0 is then based on two CFG models similar to the one used by Cohn, Blunsom & Goldwater (2010):

$$P_0(e_s, e_t|q_s, q_t) = P_{E_s}(e_s|q_s)P_{E_t}(e_t|q_t)P_{\text{alignment}}(F(e_s), F(e_t)) \tag{37}$$

where $P_{\text{alignment}}$ is the uniform distribution over all possible alignments between the frontier nodes of the source elementary tree $F(e_s)$ and those of the target elementary tree $F(e_t)$. For their application to sentence compression, they also have to handle the case where the

root state for the target tree is empty (ϵ). In this case P_0 is somewhat simplified to:

$$P_0(e_s, \epsilon | q_s, \epsilon) = P_{E_s}(e_s | q_s) \tag{38}$$

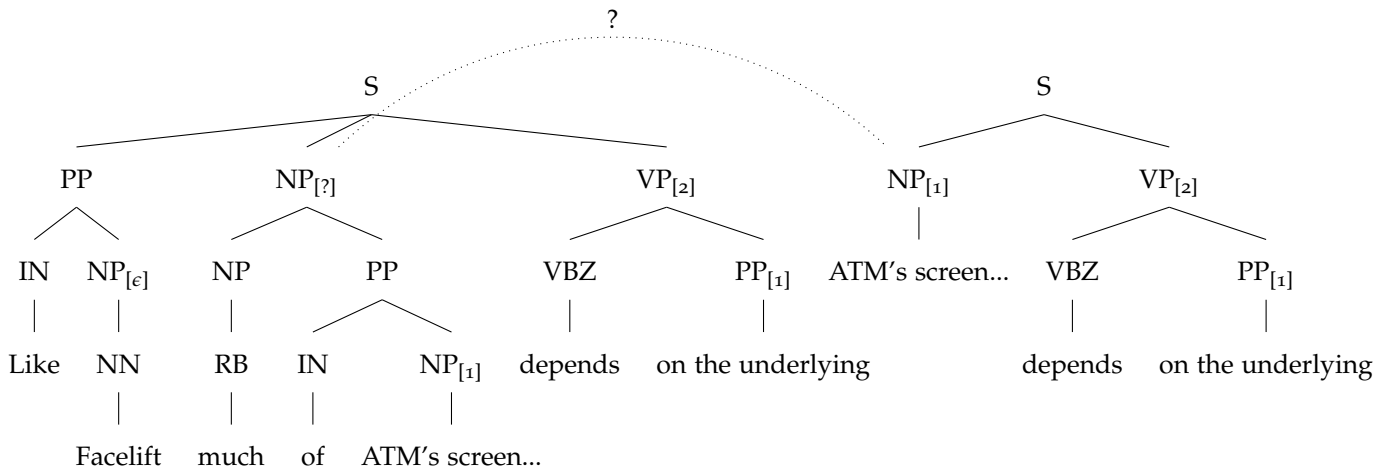
The case is similar when $q_s = \epsilon$ and $q_t \neq \epsilon$.

5.2.1 *Inducing the grammar*

So far we have seen that it is fairly simple to extend the structure of our model to handle the synchronous TSG case. Now we need to consider how to adapt the inference algorithm.

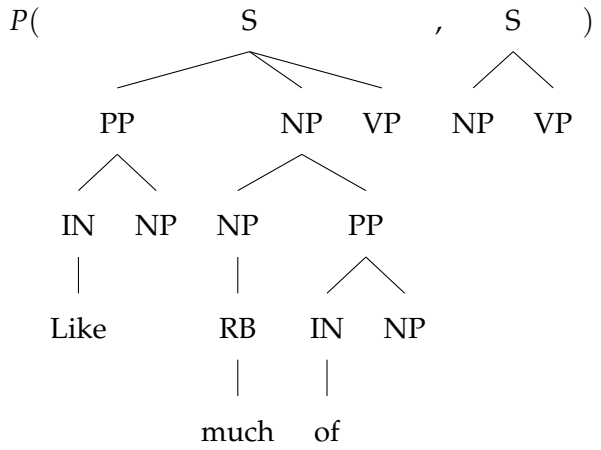
Yamangil & Shieber (2010)'s strategy is to use almost the same approach, resampling split-merge decisions at each node in the source tree. However, these decisions are now tied to the grammar of the target tree. In particular, for a node to be a substitution site in the source tree it must be aligned to some substitution site in the target tree.

Let's consider our example TreePair again, simplifying the notation to focus on one sampling decision:

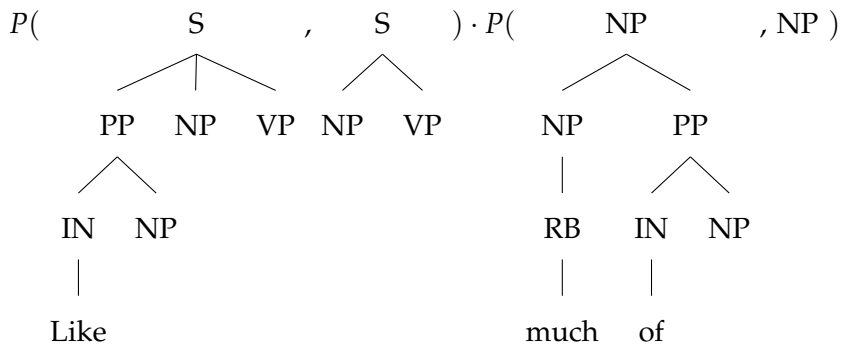


We want to figure out if this node should be (a) an interior node (and therefore unaligned); (b) aligned to the empty string (like $\text{NP}_{[\epsilon]}$); or (c) aligned to $\text{NP}_{[1]}$ (to which one of its children is already aligned). The second case (b) is trivial, because the fact that one of this nodes descendents. is already aligned to a node in the target means that it cannot align to the empty string, so this probability is 0.

If this node is unaligned (as in (a)), then we have a situation similar to our MERGE decision for the TSG: the probability of this outcome is:



Now in case (c) we consider aligning this node to the same node in the target as its descendent. The probability for this case is given by the product of two probabilities, similarly to the SPLIT for the TSG:



Again we sample the decision according to the ratio of each of these outcomes to the sum of the possible outcomes.

Yamangil & Shieber (2010) do not specify the role of alignments between frontier nodes in calculating these probabilities during sampling. In our own models, however, we specify at the top-level of our model both a CRP over possible TreePairs and a CRP over the probability of individual node states being aligned. We also will not allow many-to-one alignments of this kind, but see Chapter 9 for more details.

5.2.2 *Other work on inducing synchronous grammars*

Prior to Yamangil & Shieber (2010)'s work on sentence compression, Cohn & Blunsom (2009) and Blunsom et al. (2009) worked on models for synchronous grammars of English and Chinese for Machine Translation (MT). Cohn & Blunsom (2009) examined synchronous TSGs for tree-to-string translation using Gibbs sampling, while Blunsom et al. (2009) focused on learning latent CFG structures for string-to-string translation. The latter paper was the first to consider latent tree struc-

tures with more than one latent node label[7] and used variational Bayesian inference instead of Gibbs sampling.

Later, Xiao & Zhu (2013) followed up on synchronous TSG induction for translation between Chinese and English, focusing on the case when you have good parses for both sides of the translation during training. This work demonstrated basic inference using the Expectation-Maximization (EM) algorithm but also used variational Bayesian inference.

In addition to this work on MT, Jones, Johnson & Goldwater (2012) applied this approach to semantic parsing for the GeoQuery dataset. In their work they approached synchronous TSGs as tree transducersand used variational Bayesian inference to learn models for parsing German, Greek, English, Thai, Spanish, Japanese, and Turkish to the GeoQuery meaning representation.

5.3 CONCLUSIONS

For our own work we will define generative models over trees with labelled arcs and follow the examples highlighted here in using Gibbs sampling for inference. Unlike the synchronous TSG model described above, our approach explicitly models the both TreePairs and alignments at the top level of the model, rather than relying on the base distribution to define alignments. While we use a similar split-and-align model for inference, we also define additional 'Gibbs operators' for our models in Sections 9.3.2 and 9.3.2.

7 Prior work had always used a single node label for all latent nodes.

Part II

FRAMEWORK AND IMPLEMENTATION

The perspective and materials used in this work, we describe:

- an implementation of sentence planning using synchronous grammars,
- the datasets used to train our models, and
- our methods for evaluating the perfomance of the system.

6 FRAMEWORK AND IMPLEMENTATION

In this thesis we use statistical models to learn sentence planning rules for natural language generation. Here we elaborate upon this framework for applying machine learning to NLG and present our implementation of this framework.

This chapter builds on the background chapters and explains how the models described in Chapter 9 are implemented.

6.1 LET'S LEARN SENTENCE PLANS!

Whether manually engineering a rule-based NLG system or developing a fully end-to-end approach, we need a corpus which is representative of the texts we want to generate. For the manual approach, this will provide insight for grammar engineers and, for the automated approach, this will provide the training data for a computational model.

A corpus of texts on its own, however, is underinformative: it only represents the potential output of the system. Recent efforts to leverage large-scale language models for text generation tend to omit semantic control as a requirement for their systems: rather than trying to generate a text with a specific meaning or communicative goal, they seek to sample plausible continuations for a text which remain on-topic and in-genre (e. g. Radford et al., 2019). Assuming, however, that we want to develop a system which can express a particular meaning, we need a semantic representation for the texts in our corpus.

This semantic representation, then, must be appropriate to the domain: if it is too domain-agnostic, then we are simply shifting the problem of text generation to the level of generating the representation used as input. Designing an appropriate semantic representation is addressed in more depth in Section 7.1.1, but the key point for our framework is that the high-level document or text planning which goes into an NLG system will always require domain-specific considerations.

So developing an NLG system will always require a corpus combined with a semantic representation of its contents: this work is unavoidable. As mentioned in Section 2.3, there have been efforts to develop end-to-end statistical or neural models which can be trained directly on the semantic representations (Text Plans) and their corresponding texts (see Figure 13). We argue, however, that training end-to-end systems is underconstrained: we should leverage the task decomposition proposed in pipeline-based approaches to split this

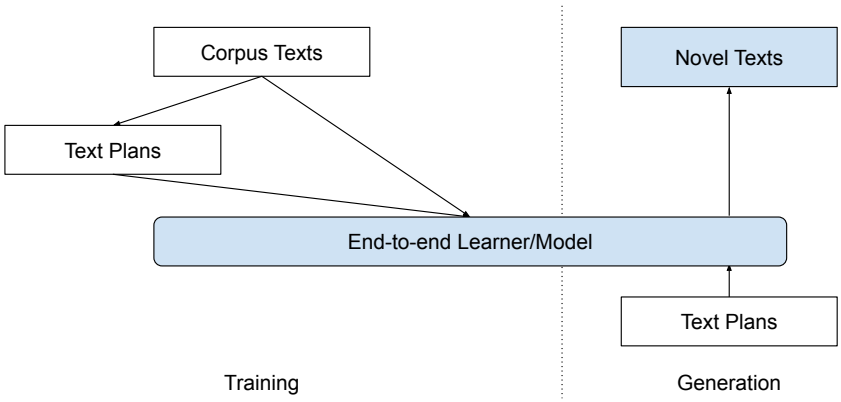


Figure 13: Training and generation pipeline for end-to-end models. Shaded blue boxes indicate the system and its outputs.

task into smaller, more learnable chunks. This argument is supported by recent work in neural NLG which found that a pipeline based approach outperformed end-to-end approaches by decomposing the task into content ordering, sentence assignment, lexicalization, referring expression generation, and surface realization (Castro Ferreira et al., 2019). Decomposing the task so that surface realization is a separate subtask also allows us to benefit from advances in (reversible) parsing and surface realization, including efforts on AMR-to-text generation and the Surface Realization Shared Tasks (Belz et al., 2011; Mille et al., 2018, 2019, 2020).

Our task decomposition is illustrated in Figure 14, with the corpus serving as input to training shown in the top left corner. We assume that the corpus includes text plans and that there is a parser which produces the same morphosyntactic representation used as input for the target surface realizer. The key component of our framework, then, is a module for learning sentence planning rules. These sentence planning rules can then be used to produce sentence plans (in the target morphosyntactic representation) for whatever (novel) text plans we wish to generate texts for. Using an off-the-shelf surface realizer, these sentence plans can then be turned into texts.

Overall, this is a simple observation: learning to map from domain-specific semantic representations to domain-general morphosyntactic representations is the task we actually need to solve automatically if we want to make it easier to develop NLG systems. In this thesis, we approach the problem as one of learning a synchronous grammar of the kind described in Chapter 3 using the statistical methods laid out in Chapters 4 and 5.

The following sections of this chapter explain the particular implementation choices made for this thesis.

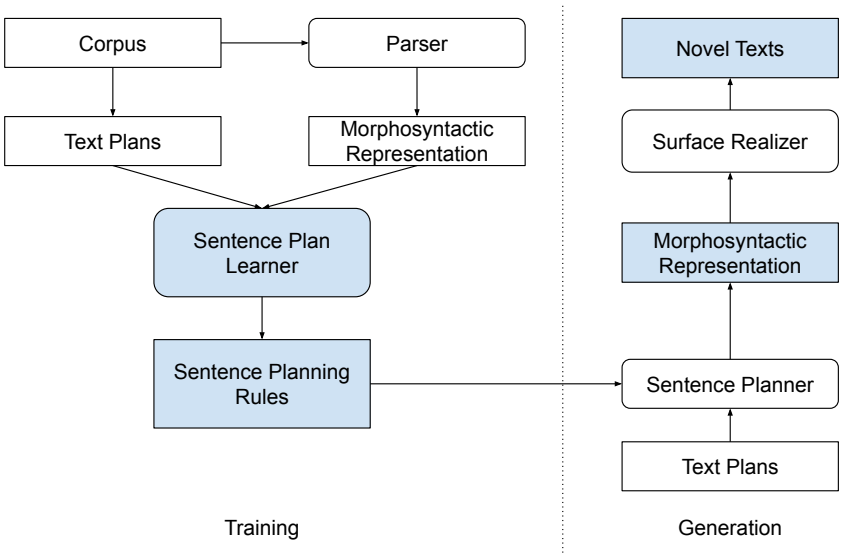


Figure 14: Our framework for training and generation, centered on learning sentence planning rules. Shaded blue boxes indicate the system and its outputs.

6.2 TECHNICAL DETAILS FOR THIS THESIS

While the models themselves are presented in Chapter 9 and the structure of our datasets is presented in Chapter 7, this section gives an overview of the tools we did not implement ourselves which were essential to the work presented in this thesis: OpenCCG for parsing and surface realization and Alto rule application. Figure 15 shows how these components fit into our implementation of the pipeline framework.

6.2.1 *Parsing and realization with OpenCCG*

We avoid the need to use separate parsers and surface realizers in this thesis by using OpenCCG, which functions as both a parser and a surface realizer (Baldridge & Kruijff, 2003; White, 2004). While OpenCCG uses Combinatorial Categorial Grammar (CCG) (Steedman & Baldridge, 2006) to represent the syntactic constraints for both parsing and realisation, our work leverages the so-called Logical Forms (LFs) that OpenCCG uses to represent the semantic content of a text rather than manipulating the syntactic CCG derivation trees.

OpenCCG comes with scripts for extracting a broad-coverage grammar of English from CCGbank (Hockenmaier & Steedman, 2007, itself based on the Penn Treebank). Using this broad-coverage grammar, we can parse the texts in our corpora into LFs which, in the case of the broad-coverage grammar, are more syntactic than semantic in nature: they strongly resemble dependency grammar trees with mor-

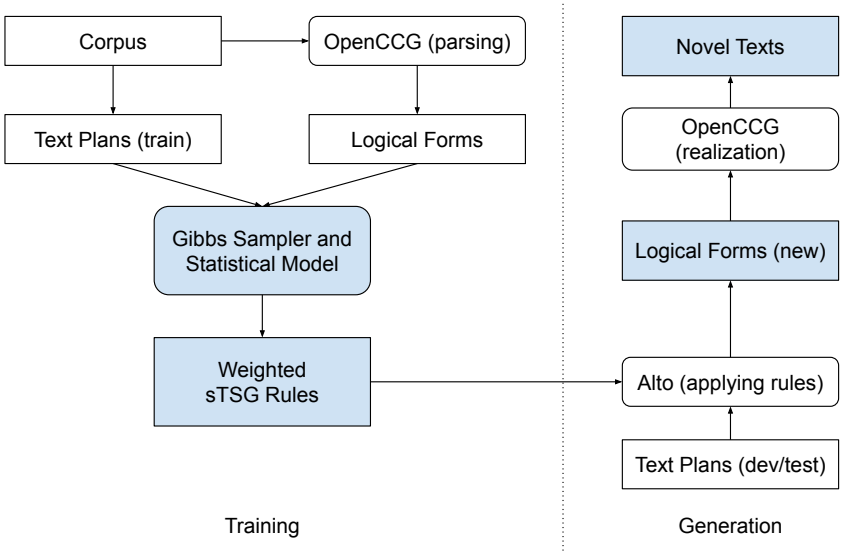


Figure 15: Our implementation of the framework. Shaded blue boxes indicate the system and its outputs.

phosyntactic annotations. Figure 16 shows two such LFs based on the broad-coverage grammar which have been joined together.

While OpenCCG can be used with grammars capable of parsing multi-sentence texts (Nakatsu & White, 2010), these grammars did not work well on our corpora in practice. Therefore we split each text into a sequence of sentences and parsed each sentence separately with OpenCCG. Our synchronous TSG approach to sentence planning rules, however, requires pairs of text plans (TPs) and logical forms which are singly-rooted trees: we cannot directly pair a text plan with a sequence of trees for our models. In order to combine the LFs for a text into a single tree, we create a right-binary-branching structure over the LFs, with the simplest case shown in Figure 16. This allows our sTSG model to learn rules which span multiple sentences.

Once training the sTSG model is complete, we have a set of sentence planning rules which can apply to a TP to produce a novel LF by using Alto (explained in Section 6.2.2). Because the model is trained on multisentence LFs, the resulting LF can also contain multiple sentences. Therefore we need to extract the sentence-level LFs before we can perform surface realization with OpenCCG. Fortunately, we can do this simply by splitting at any *SEQ* nodes appearing in the LF (cf. Figure 16). The resulting sequence of LFs for a given TP can then be realized one at a time by OpenCCG, and the resulting strings can be concatenated to form a text for the input TP.

Strictly speaking, OpenCCG does not require its LFs to be tree-structured: they can contain `idref` nodes which simply point to another part of the tree. This is a convenient way of encoding, e.g., the subject of a relative clause, as shown in Figure 17. For our purposes,

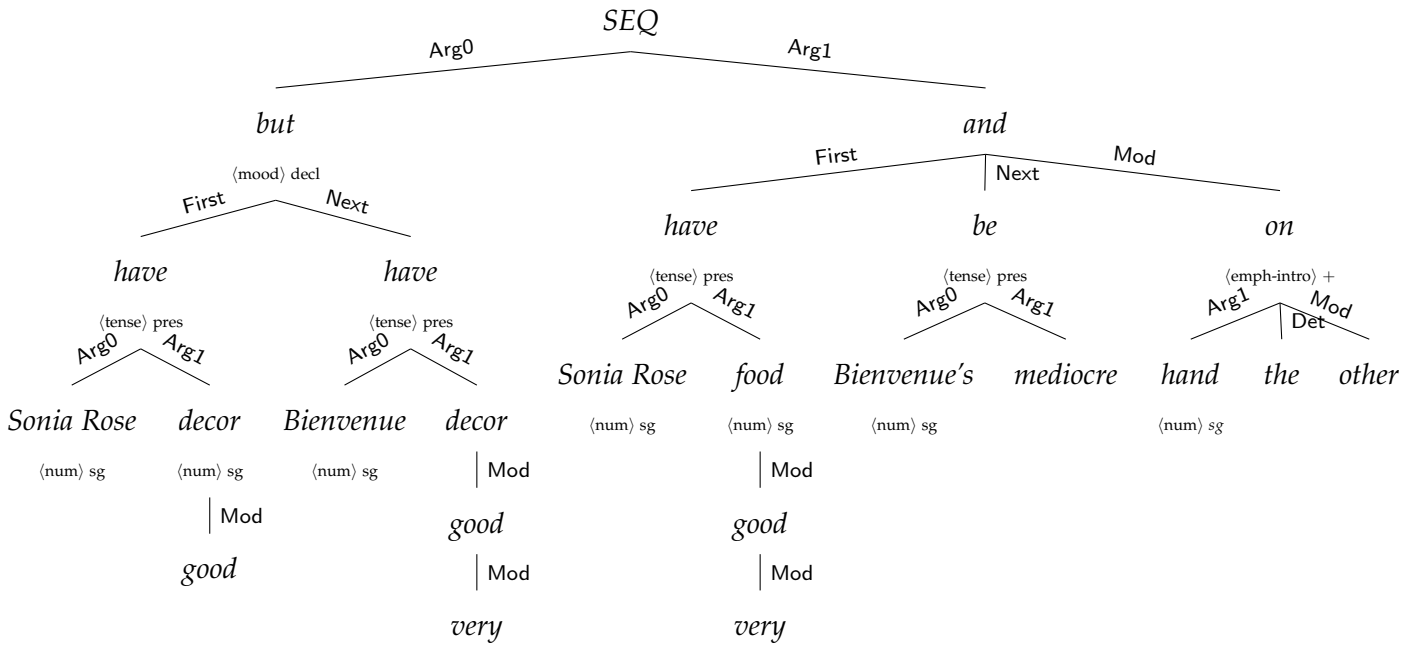


Figure 16: Multi-sentence logical form for the text 'Sonia Rose has good decor, but Bienvenue has very good decor. On the other hand, Sonia Rose has very good food and Bienvenue's is mediocre.' The *SEQ* node is inserted to join two sentential LFs together. For three or more sentences, we simply nest binary *SEQ* nodes.

however, we can treat these idref nodes as leaves in a tree-structured LF during sTSG induction.

6.2.2 *Rule application with Alto*

Alto (Koller & Kuhlmann, 2012) provides an efficient implementation of parsing for synchronous grammars. We use this parsing implementation to identify derivations for the text plans in our corpus and expand the corresponding logical form derivations into a complete LF. In order to use Alto in this way, we convert the labelled arcs in our TPs and LFs into intermediate unary nodes, as illustrated in Figure 18.

Internally Alto uses the weights assigned to each sentence planning rule by our statistical models to determine the k-best parses with the Viterbi algorithm. After Alto produces a k-best list of LFs for each Text Plan (TP) in a run, we realize all of these with OpenCCG and then re-rank them, as detailed further in Section 9.6.1.

6.2.3 *Implementing our models in Python*

All of the code for this thesis is implemented in Python 3.6. Using a dynamically-typed language which supports type annotations for

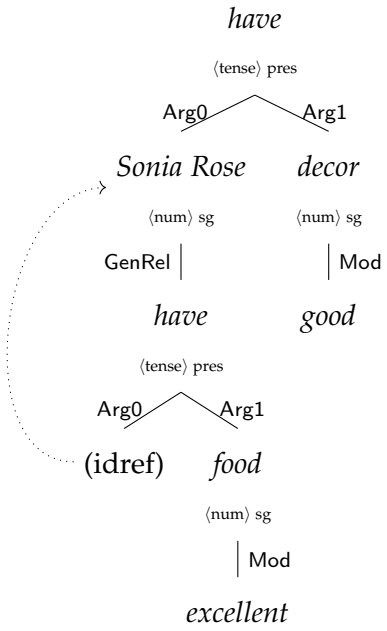


Figure 17: An LF representing the sentence 'Sonia Rose, which has excellent food, has good decor'. Note the use of an (idref) node, which simply serves to point to the node which fulfills the Arg0 role for the second *have* node.

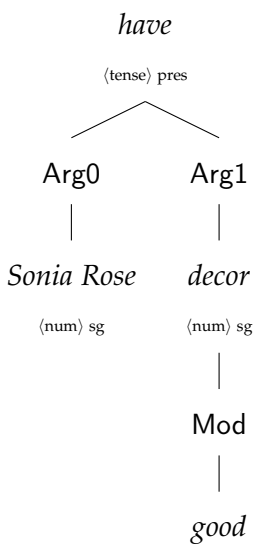


Figure 18: An LF representing the sentence 'Sonia Rose has good decor' where we have converted labelled arcs into nodes bearing the arc label with a unary expansion.

static analyses facilitated rapid development and prototyping while preserving the ability to document expected datastructures to avoid programmer errors once the code enters into regular usage.

We created two modules to separate the manipulation and representation of linguistic structures and file types (in `lingstruct`) from the representation and training of statistical models (in `bn4nlg`). The former contains code for manipulating file types used by OpenCCG and Alto as well as Python data structures for representing different kinds of text plans and logical forms. The latter includes our implementation of the Chinese Restaurant Process (Section 4.4), our Gibbs samplers (Section 9.3), and classes representing the (different components of our) statistical models (Section 9.1). `bn4nlg` includes bash scripts and a Makefile for running OpenCCG, Alto, and a baseline neural model (TGen) while `lingstruct` includes a number of Python command-line tools for manipulating different filetypes.

6.3 CONCLUSION

In this chapter we have argued for the importance of machine learning for sentence planning and highlighted the tools used in our own implementation of this framework. Focusing on sentence planning allows us to make minimal manual efforts in system development, to leverage advances in surface realization, and to have an interpretable set of rules determining the capabilities of our NLG system.

7 CORPORA

Based on the framework for training Natural Language Generation (NLG) systems developed in the previous chapter, we now shift our focus to identifying the necessary properties of a corpus for training in this framework. On this basis we can then evaluate existing resources & and motivate the collection of a new corpus. Finally we present the Extended SPaRKy Restaurant Corpus (ESRC), which we built in order to fill the gap present in existing resources.

The SPaRKy Restaurant Corpus (SRC) and ESRC are used to training and evaluate the synchronous grammar models we develop in Chapter 9.

7.1 NECESSARY PROPERTIES

A corpus for training an NLG system has two basic requirements: an adequate semantic representation and a level of variation commensurate with the desired output.

7.1.1 *Concise Semantics & Hierarchical Discourse Structure*

Focusing on sentence planning in this thesis, we assume that content selection and document planning are handled by another system. Our goal is a system which reduces the amount of human effort required to start generating in a new domain. Therefore we focus on text plans which are easy for a domain expert to write given some corpus text.

In particular, we want to collapse distinctions across different linguistic forms used to convey the same meaning in the target domain. For example, the sentences 'Sonia Rose serves nice food', 'The food is lovely at Sonia Rose', and 'Sonia Rose has good food quality' should all have the same semantic specification in the restaurant recommendation domain, despite the fact that there are linguistically interesting semantic and pragmatic differences between them. For these sentences, we might use FoodQuality(SR, good) as the semantic representation, also known as a Meaning Representation (MR).

Collapsing subtle linguistic distinctions in this way also serves to ensure that a system trained on such a corpus can learn lexicosyntactic variants for each meaning it must communicate. This stands in contrast to the representations commonly used for surface realizers, which include information about the specific lexical choices (e.g. *lovely* versus *nice*) and their syntactic relationships to one another: in order to learn to generate varied texts, we must have a one-to-many

mapping between semantic representations and their linguistic realisations.

The Cambridge University Engineering Department Standard Dialogue Acts (Young, 2009) represent one common approach to these concise semantic representations. Each CUED dialogue act includes a dialogue act (e.g. INFORM, REQUEST, CONFIRM) and (optionally) a set of slot-value pairs representing propositions to be conveyed to a user (or a system). For example, (1) represents an informative dialogue act communicating the name of an establishment and the quality of its decor. While the CUED format allows for dialogue acts like `hello()` for generating a greeting message and `reject(slot1=value1)` for stating that a particular slot-value pair is not a possible option, corpora for NLG focus almost exclusively on the INFORM dialogue act. (2) presents the same information, but respresented as a conjunction of two separate dialogue acts.

(1) `inform(name=Sonia_Rose;decor=good)`

(2) `inform(name=Sonia_Rose), inform(decor=good)`

These concise semantic units work well when we need to generate short descriptions of a single entity, but it does not provide a way to represent the discourse relations we may want to communicate to users. Continuing our example from the restaurant domain, if we want to produce simple reviews we do not necessarily need to express CONTRAST relations between two propositions; we can simply describe the food quality and decor at the restaurant directly. If, however, we want to compare restaurants to one another, we may want to highlight their differences.

Consider again Figure 6, reproduced here as Figure 19. The hierarchical structure of Figure 19 is more expressive than the corresponding CUED dialogue acts in Examples 3 and 4: the dialogue act representation can potentially be used to represent sentence or clause boundaries, but does not provide any way to highlight the relationships between particular propositions (CONTRAST in this case).

(3)
```
inform(name=JP;cuisine=Italian,Pizza),
inform(name=CBG;cuisine=Italian),
inform(name=JP;price=20),
inform(name=CBG;price=26),
inform(name=JP;food_quality=very_good),
inform(name=CBG;food_quality=good)
```

(4)
```
inform(name=JP;cuisine=Italian,Pizza;price=20;food_quality=very_good),
inform(name=CBG;cuisine=Italian;price=26;food_quality=good)
```

Naturally the level of discourse representation required will also depend on the length of the target text. For simple reporting of facts in one or two sentences, it may suffice to leave the discourse structure underspecified and simply lexicalize a sequence of facts, using a

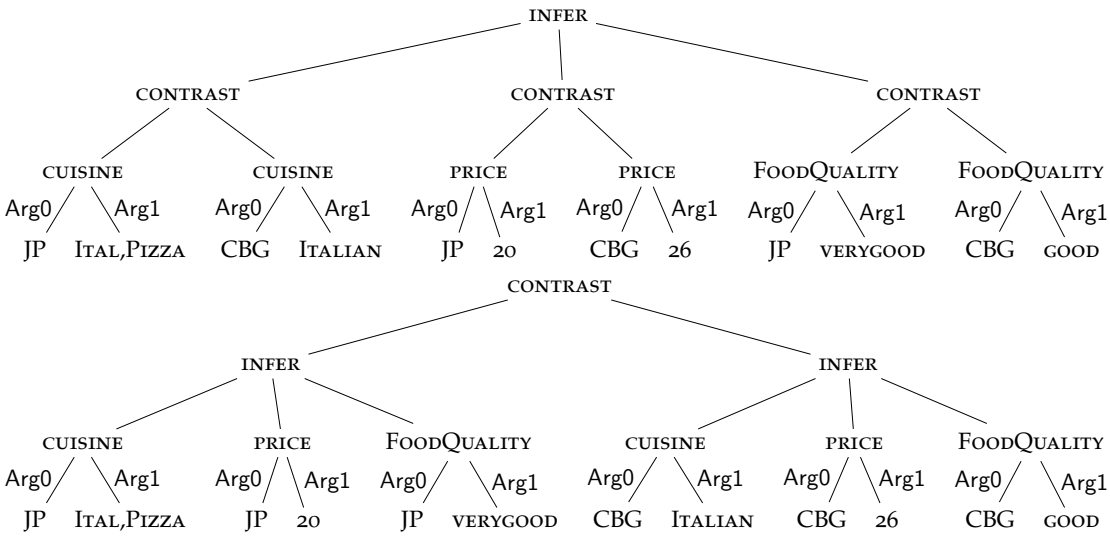


Figure 19: Two possible textplans encoding specific strategies for expressing the contrasts shown in Figure 5. For discussion of these different strategies and an explanation of the INFER relation, see Section 7.3.2.1.

semantic representation more like the CUED dialogue acts. If, on the other hand, the content to be communicated is a paragraph or longer in length, it is unlikely that such a simple discourse representation will adequately capture the relations to be communicated.

7.1.2 *Appropriate Variation*

When hand-crafting a rule-based system we can rely on the expertise and linguistic experience of our engineers to provide additional creativity beyond the example texts they are given, but when we want to train an NLG system automatically, we cannot rely on this creativity.

Instead, all of the variation we want our system to be able to express must be present in the training texts. For example, if we wish to create systems which can adapt the linguistic complexity of their texts to different users and different situations, it is necessary to ensure that our training texts include a range of texts appropriate to these different use cases.

Evidence from psycholinguistics suggests that a text's *information density* (Shannon & Weaver, 1948, also known as *surprisal*) is one relevant measure of its linguistic complexity, both with respect to reading (Demberg & Keller, 2008) and language production (Asr, 2015; Jurafsky et al., 2001; Raymond, Dautricourt & Hume, 2006). The fact that human language production is sometimes sensitive to this measure is especially relevant to NLG efforts aiming to be human-like. For our purposes, then, it is important to identify or create a corpus which exhibits differences with respect to the information density of its texts.

information density

Table 6: Lexical variation and average text length across corpora in the restaurant domain. 'Vocab' is the number of words occurring at least 5 times in the corpus.

			mean text length (range, sd)		MR		
corpus	texts	vocab	words	sentences	tokens	types	delex.
BAGEL	404	74	11.55 (4-26, 3.45)	1.03 (1-2, 0.17)	404	381	202
SF Rest.	5192	353	9.00 (1-35, 5.30)	1.05 (1-4, 0.25)	5192	1950	217
E2E	51426	990	20.34 (1-71, 6.93)	1.56 (1-6, 0.71)	51426	6039	120
SRC	1760	99	39.75 (10-160, 25.50)	4.27 (1-25, 3.33)	1760	77	52

7.2 EXISTING CORPORA FOR DATA-DRIVEN NLG

Existing resources either (1) use a flat MR or (2) pair hierarchical MRs with texts produced by a traditional, hand-crafted NLG system.

This section describes existing corpora which use the CUED dialogue act meaning representation. The next section then describes the only publicly-available corpus for NLG to use a richer semantic representation prior to our work: the SPaRKy Restaurant Corpus.

Table 6 offers summary statistics for the corpora under consideration while Table 7 shows example MRs and texts for each corpus.

7.2.1 *BAGEL*

The BAGEL corpus (Mairesse et al., 2010) consists of 202 CUED-style dialogue acts, each of which has two texts, resulting in a corpus of 404 MR-text pairs. These texts were manually aligned to the slot-value pairs of the meaning representations and used to train a dynamic Bayesian network for generation.

To facilitate alignment, the BAGEL corpus explicitly breaks down a single dialogue act (as in Example 1) above into a sequence of acts expressing individual attribute-value pairs (as in Example 2). The manual alignments between these component parts and words in the associated texts is indicated in their encoding of the text (as shown in Table 6).

7.2.2 *Wen et al. corpora*

Wen et al. (2016; 2015b) collected four datasets for training neural NLG models in four domains: restaurant search, hotel search, laptop sales & search, and television sales & search. These corpora also use the CUED format but do not include manual alignments between the parts of the MR and parts of the texts. However, these corpora are each an order of magnitude larger than the BAGEL corpus. Table 6

Table 7: Example MRs and texts from NLG corpora in the restaurant domain. BAGEL includes the raw MR as well as an abstract MR which has been delexicalized, as well as alignments between (sequences of) words and the delexicalized MR. E2E only uses the INFORM dialogue act, so it is omitted from the MR. SRC includes hierarchical structure, which we represent here in a slightly simplified form along with 2 of the 20 texts provided for this MR.

	MR
corpus	Text
BAGEL	`inform(name="Ali Baba", type=placetoeat, eattype=restaurant, area=riverside, near="The Bakers", near="Avalon")`
	`inform(name="X1", type=placetoeat, eattype=restaurant, area=riverside, near="X2", near="X3")`
	[near]Close to [near]both the [near+X]X [near]and [near+X]X []you will find the [area+riverside]riverside [eattype+restaurant]restaurant, []The [name+X]X
SF Rest.	`inform(name='trattoria contadina'; price_range=moderate)`
	trattoria contadina is a nice place it is in the moderate price range
E2E	`name[The Vaults], eatType[pub], priceRange[more than £30], customer rating[5 out of 5], near[Café Adriatic]`
	The Vaults pub near Café Adriatic has a 5 star rating. Prices start at £30.
SRC	`contrast( nucleus:CUISINE(<Caffe Buon Gusto>;<Italian>), nucleus:CUISINE(<John's Pizzeria>,<Italian Pizza>))`
	Caffe Buon Gusto is an Italian restaurant while John's Pizzeria is an Italian , Pizza restaurant.
	...
	Caffe Buon Gusto is an Italian restaurant. On the other hand, John's Pizzeria is an Italian , Pizza restaurant.

Figure 20: Example image from (Novikova, Lemon & Rieser, 2016) indicating an expensive Italian restaurant named the Wrestlers with high ratings but not family-friendly near the river and a place called Cafe Adriatic.

includes summary statistics for the restaurant-domain corpus from Wen et al. (2015b, SF Rest.), since this domain has received the most attention to date and is also used in this thesis.

7.2.3 *End-to-End Generation Challenge*

The End-to-End Generation Challenge dataset (Novikova, Dušek & Rieser, 2017, E2E) is also in the restaurant domain and provides another order-of-magnitude improvement over the individual Wen et al. corpora, containing $> 50k$ texts. While both the BAGEL and the Wen et al. corpora were elicited by presenting human subjects with meaning representations and asking them to produce a text communicating that content, the E2E dataset follows Novikova, Lemon & Rieser (2016) in using images to elicit texts (Figure 20). Using images instead of dialogue act MRs to prompt subjects meant that they were less likely to repeat exactly the phrases used in the MRs, resulting in a wider variety of lexicalizations compared to the earlier corpora. Similarly, images do not provide the same rhetorical or discursive framing, allowing participants a greater sense of freedom to order the facts as they see fit.

While the goal in designing the corpus was for participants to express all of the content given in the infobox (the white rectangle), no restrictions were put in place to ensure that they did so. This means

that any system trained on the original E2E dataset had to learn content selection as well as sentence planning and surface realization. More recently we collaborated with the original authors of the dataset in order to demonstrate the impact of this semantic mismatch (i.e. the difficulty of jointly learning content selection and sentence planning & surface realization) and released a cleaned version of the dataset (Dušek, Howcroft & Rieser, 2019).

7.3 THE SPARKY RESTAURANT CORPUS

The SPaRKy Restaurant Corpus (Walker et al., 2007, SRC) provides a richer semantic representation more appropriate to the kinds of texts we would like to generate.

This dataset served as source material for the development of a more varied corpus of restaurant recommendations and comparisons, as discussed in Section 7.4. While this corpus does not exhibit full range of lexical, syntactic, or information density variation that we would like our system to express, our evaluations (Chapter 9) use models trained on the SRC provide a sanity check that our system is able to recapitulate a simple hand-crafted sentence planner.

7.3.1 *Background*

Walker et al. (2007) developed the Sentence Planner with Rhetorical Knowledge (SPaRKy) as a replacement for a template-based generation system in the context of the Multimodal Access to City Help (Walker et al., 2004, MATCH) project.In this work Walker et al. developed a multimodal interface to provide information about restaurants in New York City. Users were tasked with finding restaurants in a particular price range or serving a particular cuisine in a given neighborhood. The texts generated by their system in the course of this task form the corpus we refer to as the SRC.

The corpus contains unordered text plans representing collections of facts to communicate and the discourse relations between them, drawn from a subset of the relations in Rhetorical Structure Theory(Mann & Thompson, 1988, RST). The corpus also contains *ordered* text plans (in which the RST relations have been encoded in a tree-structure), the sentence plans passed to the surface realizer, and the output texts.

For our purposes, the important element is the ordered text plans, so we will briefly describe the structure of these text plans, including which Rhetorical Structure Theory (RST) relations and propositions are used. We also summarize the clause-combining operations present in the SRC to highlight the simplicity of the rules present in this corpus.

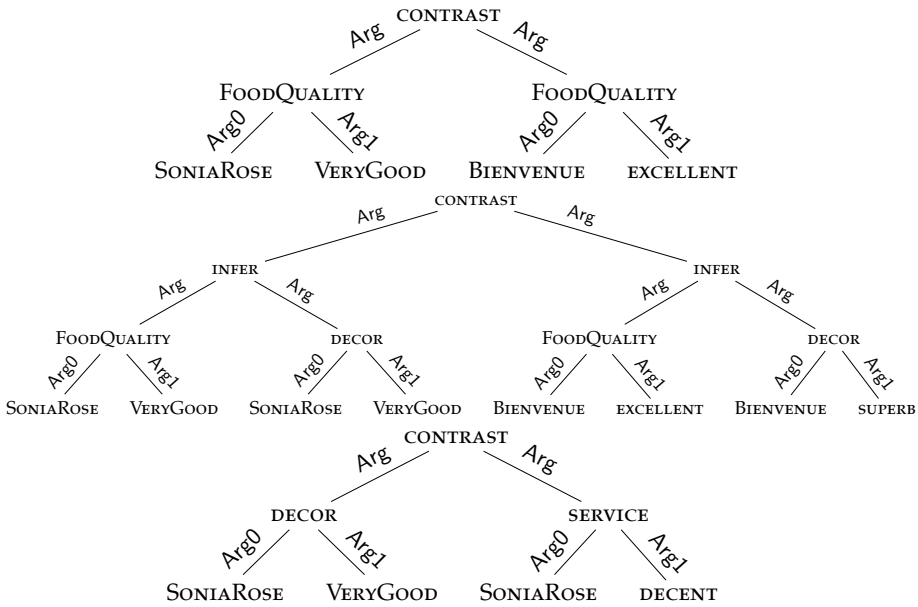


Figure 21: Three examples of the contrast relation in the SRC domain. The top two trees can appear in the SRC, while the final tree represents an enhancement suggested in Howcroft, Nakatsu & White (2013).

7.3.2 *Ordered Text Plans*

7.3.2.1 *Rhetorical Relations*

While Mann & Thompson (1988) present 23 different relations for describing the rhetorical structure of a text, the SRC uses only three of these for its text plans: contrast, justify, and elaborate. In addition to these, the INFER relation simple clusters propositions together, juxtaposing them and directing the sentence planner to leave the relation between them implicit for the reader.

The CONTRAST relation connects two propositions which are mostly similar in order to draw attention to the ways in which they differ. In the SRC this relation is only used to express contrast between the properties of two restaurants, either comparing only a single property which differs between the two (as in first tree in Figure 21) or comparing a cluster of properties which differ (as in the second tree in 21). While in earlier work we have proposed enhancing the expression of contrast in this corpus by inserting CONTRAST relations between the properties associated with a single restaurant (as in the final tree in Figure 21) (Howcroft, Nakatsu & White, 2013), we use the corpus in its original form for the experiments presented in this thesis.

The JUSTIFY relation appears in two varieties in the SRC: nucleus-first and satellite-first. The *nucleus* is the main argument of the relation, in this case the proposition whose claim is supported by the *satellite* argument(s). In our representation of the corpus we only allow INFER nodes to have more than two children, so if there is more

than one satellite argument to a justify relation, these arguments are first grouped under an INFER relation.

Finally, the SRC includes the ELABORATE relation, which behaves similarly to the JUSTIFY relation. For this relation, the nucleus always comes first and the satellite nodes are not evidence for the nuclear proposition. Instead they are used to provide additional detail about the situation or some element of the nuclear proposition.[1]

7.3.2.2 *Propositions*

The SRC contains two simple attributive semantic predicates: BEST for specifying that a single restaurant is the best among a selection and LISTEXCEPTIONAL for listing a set of restaurants all of which offer 'exceptional' value. Of these, the latter appears only in the text plans comparing three or more restaurants.

The remaining predicates take two arguments: a restaurant name and a value. The PRICE predicate describes the average cost of a meal at the restaurant while NEIGHBORHOOD describes the area in which the restaurant is located. Finally there are the FOODQUALITY, DECOR, and SERVICE predicates for describing the quality of these aspects of the restaurant. These predicates all take 'scalar' adjectives as their arguments (e.g. 'mediocre', 'decent', 'excellent').

7.3.3 *Clause-Combining Operations*

The SRC contains six clause combining operations, ranging from simple sentence concatenation (**period**) to replacing full sentences with a simple prepositional phrase (**with-reduction**).

- **Period** simply concatenates two sentences and is named for the punctuation mark which comes between them.
- **Merge**, or object coordination, allows for NP coordination when the matrix verbs of two adjacent clauses are the same and all but one of their arguments are identical.
- **With-reduction** can reduce the FOODQUALITY, DECOR, and SERVICE propositions to a simple prepositional phrase beginning with 'with' if the proposition is adjacent to another proposition about the same subject.
- **Relative clause** allows two adjacent clauses with identical subjects to be combined by making one of the two a subject relative clause embedded in the other.

1 Mann & Thompson (1988) describe six possible scenarios for using the ELABORATE relation: relating sets and their members, abstract concepts to instantiations thereof, wholes and their constituent parts, processes and their individual steps, objects and their attributes, and generalizations to specific examples thereof.

- **Cue word conjunction** combines two clauses by subordinating one to the other with a conjunction, while **cue word insertion** behaves similarly to the **period** operation with the exception of the insertion of a discourse cue word at the beginning of the second sentence.

7.4 THE EXTENDED SPARKY RESTAURANT CORPUS

While the SPaRKy Restaurant Corpus (SRC) has the desired sort of semantic representation, we have seen that its lexical and syntactic variation are quite limited. This limited variation means that a system which manages to learn sentence planning rules on the SRC will not necessarily work when applied to more natural training data, where propositions and discourse relations are less likely to be systematically expressed in the same way. Of course, this limitation also restricts the amount of variation possible in the resulting systems trained on the SRC.

The Extended SPaRKy Restaurant Corpus (Howcroft, Klakow & Demberg, 2017, ESRC) represents a more realistic dataset for training an NLG system and captures a wider variety of lexical and syntactic choices than the SRC. Moreover, we designed this corpus so that it exhibits variation specifically with respect to *information density* to facilitate the creation of an adaptive generation system.

7.4.1 *Corpus development*

In developing this corpus we sought to provide texts which:

1. use the same rich semantic representation present in the SRC (per Sec. 7.1.1);
2. vary with respect to information density (per Sec. 7.1.2); and
3. are more representative of the kind of corpus system developers might create or find quickly and easily.

To this end, we crowdsourced paraphrases of SRC texts using two sets of instructions in order to elicit texts which differ with respect to average information density. Based on the text plans of the original texts, we then manually corrected the semantic annotations for a subset of the new corpus.

7.4.1.1 *Experiment design*

At first glance, creating a corpus of natural language which is more representative of human language use than a corpus consisting of the outputs of a rule-based NLG system seems straightforward: nearly any corpus in our target domain written by humans should suffice.

However, recall the problems in the BAGEL and Wen et al. corpora as compared to the E2E corpus: these texts did not elicit much lexical variation, presumably because the subjects writing the texts were looking at a text-based meaning representation of the slot-value pairs they needed to communicate.

Unlike Novikova, Dušek & Rieser (2017) and Novikova, Lemon & Rieser (2016), we would like to create texts with a specific hierarchical meaning representation (i.e. the discourse structure specified in our text plans).Therefore it does not make sense to use a field of images presented to users in order to elicit texts, as this format does not allow us to highlight the contrasts or justifications we would like the subjects to describe.

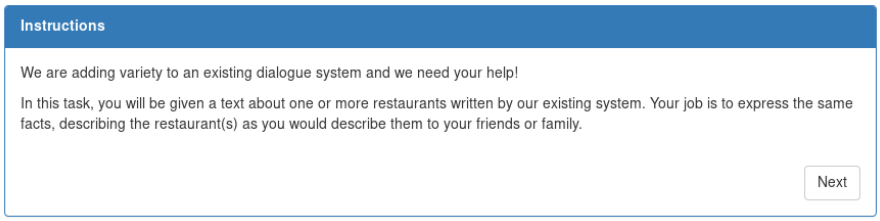
Instead we use a paraphrasing paradigm based on the original SRC texts.Using these texts strikes a balance between these two approaches with respect to lexical variation, because slot-value pairs are still encoded textually but they are less clearly demarcated when they are a part of running text. This also means that subjects will read explicit discourse cues and are therefore more likely to include the discourse relations in their own rephrasings.

In order to collect these paraphrases, we created a LingoTurk (Pusse, Sayeed & Demberg, 2016) template for elicitation. LingoTurk is a framework for crowdsourced (psycho)linguistic experimentation which makes it easy for researchers to run experiments on different crowdsourcing platforms, ensures that each subject sees only as many lists as they are allowed to, and takes care of item randomization. Several screenshots of our template are shown in Figure 22.

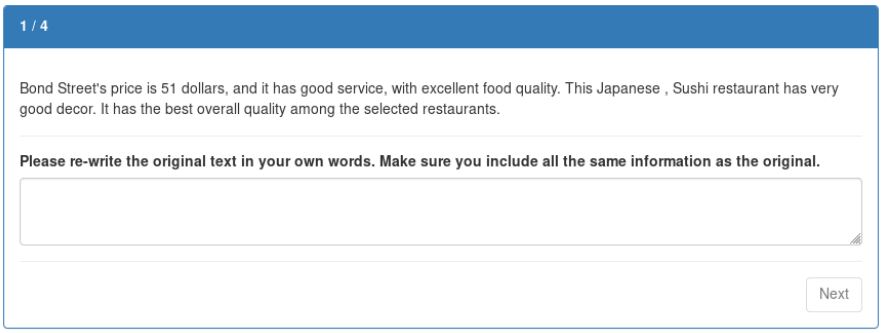
The difference between the instructions in the DEFAULT and ELDERLY conditions is clear in Figure 22. In the DEFAULT condition subjects simply had to rewrite the text for a familiar audience, while in the ELDERLY condition they were instructed specifically to describe the restaurants to their '85-year-old' grandmother. Based on the phenomenon of elder-speak, we expected subjects to produce utterances with lower information density when paraphrasing a text for an elderly grandparent (the ELDERLY condition) compared to the DEFAULT condition of paraphrasing a text for their friends and family in general. While other target listeners could be used to elicit information with a reduced information density (e.g. a child listener), such listeners are not expected to need the same information as an adult listener to aid their decision making when choosing a restaurant to dine at. In talking to an elderly relative, however, we expected that our participants would produce relatively standard adult-directed language while compensating for stereotyped cognitive decline.

elderspeak: the tendency of speakers to 'simplify' their language when speaking to the elderly

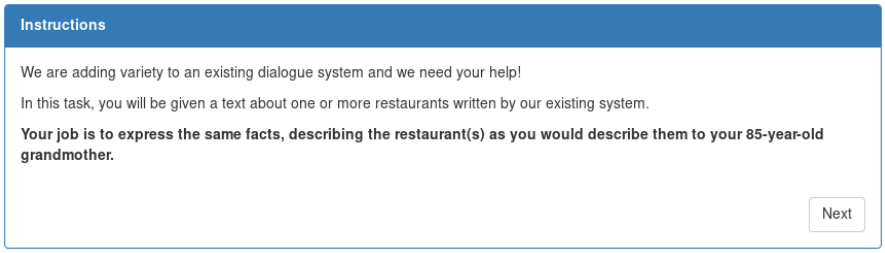
Each participant saw only one condition (DEFAULT or ELDERLY) and paraphrased two recommendations of a single restaurant and two



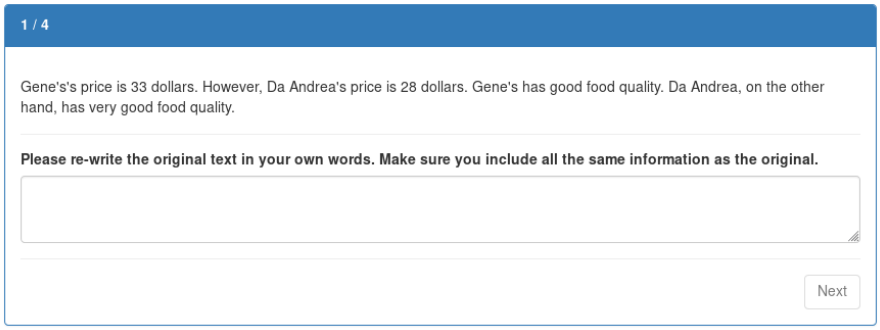
(a) Instructions in the DEFAULT condition.



(b) Sample prompt in the DEFAULT condition.



(c) Instructions in the ELDERLY condition.



(d) Sample prompt in the ELDERLY condition.

Figure 22: Instructions and elicitation screens for the DEFAULT and ELDERLY conditions of the experiment.

comparisons of two restaurants.[2] This took about 7 minutes on average, and we paid subjects 1 GBP each for their participation. We required participants to be native speakers of English living in English-speaking countries.

Each of the 672 randomly selected SRC texts used as paraphrasing prompts was paraphrased at least 4 times in each condition, yielding more than 2600 texts in each condition.

7.4.1.2 *Text cleanup*

Given the importance of good text quality for training inputs to an NLG system, we processed these texts to normalize spelling and restaurant name mentions (e. g., correcting 'restuarant', ensuring the restaurant 'Il Mulino' was not abbreviated to 'Mulino'). We also ensured that every sentence begins with a capital letter and ends with a punctuation mark. For the 1344 texts we manually annotated, we further corrected uses of nonstandard sentence-final punctuation (e. g., 'run-on' sentences using commas in place of full stops).

These corrections taken together affected almost three-quarters of the texts (in contrast to the 7% of items requiring spelling correction reported by Novikova, Lemon & Rieser (2016)).These corrections included adding 882 sentence-final periods, with this being the only change in 514 of the texts, as well as correcting the restaurant name in 987 sentences.

7.4.1.3 *Semantic annotation cleanup*

Because every text in the ESRC is based on a text in the SRC, we have an initial text plan for each text. However, it was not uncommon for participants to rewrite the source texts in a way that changed either the discourse structure of the text or the set of propositions in the text. In order to ensure that the corpus includes high quality semantic annotations, we manually corrected one quarter of the text plans in the corpus. This effort makes it possible to explore a staged learning approach using first the highly constrained SRC to initialize a model, then adding in texts from the manually verified subset of the ESRC, before ultimately incorporating texts from the rest of the ESRC.

We used a custom commandline script to first view each text along with the set of propositions intended to be in that text. This tool made it easy to re-order the propositions and to change their values to align well with the text. In most cases this was fairly straightforward, however there were some texts where a proposition was expressed in a

2 Our choice to limit the comparisons to two restaurants is guided by Howcroft et al.'s (Howcroft, Nakatsu & White, 2013) finding that the comparisons of three or more restaurants in the SRC were too complex to be rated highly by human subjects. This also results in the exclusion of one proposition type from our dataset: ListExceptional.

discontinuous way, such as *Lovely decor at Sonia Rose which is simply beautiful!*

After this initial correction of the semantic content of the annotations, we used a second script to view the texts and the lists of propositions in order to assign a discourse structure to the text.

Validating the approximately 6k propositions in the original texts for these 1344 paraphrases and completing the other corrections referred to in Section 7.4.1.2 took about 40 person-hours of work. During this process, we codified the criteria used to correct and annotate the texts in a set of guidelines which were then validated through feedback from colleagues.

Of these 6k propositions, subjects altered 580 by, for example, describing a restaurant's decor as 'excellent' when the original text merely described it as 'good'. This is one of the risks in crowdsourcing data, where subjects are used to industrial tasks where they are supposed to write positive reviews and are therefore likely to exaggerate relative to the intended meaning. Of course, when developing an NLG system we need to remove such noise to ensure that the resulting NLG system is truthful.In this example and other similar cases, our approach was to alter the text plan to match the actual text provided by the user, which has the effect of increasing the number of different text plans we have in our corpus.

Participants also completely dropped about 320 propositions in their texts, or about 5 percent of the total propositions in the original texts. Even combined with the 580 propositions whose values were altered, this represents only a 15% rate of omission (cf. the 22% reported in Novikova, Dušek & Rieser, 2017). The majority of the paraphrases, however, were not affected by these alterations, with 830 of the 1344 we annotated preserving all of the original content (i.e. 61.8%, quite similar to the 60% reported in (Novikova, Dušek & Rieser, 2017)).

7.4.2 *Statistics*

Our corpus consists of more than 5300 texts along with meaning representations based on the paraphrased original SRC source and manually quality-checked and corrected meaning representations for 1344 of these. The corpus exhibits a wide range of lexical variation and variation with respect to information density, word & sentence length, and proposition density.

7.4.2.1 *Lexical variation*

Our corpus contains more than 1500 unique words, more than 500 of which occur 5 or more times. This is a marked improvement of the $\approx$ 65 words occurring in the portion of the original SRC which we used to as prompts in our paraphrasing task. Consider, for example,

Table 8: Lexical variation across corpora in the restaurant domain. 'Vocab' is the number of words occurring at least 5 times in the corpus. 'Sem. ESRC' is the subset of the data with manually corrected semantics. 'Full ESRC' is our corpus of paraphrases.

			mean text length (range, sd)		MR		
corpus	texts	vocab	words	sentences	tokens	types	delex.
BAGEL	404	74	11.55 (4-26, 3.45)	1.03 (1-2, 0.17)	404	381	202
SF Rest.	5192	353	9.00 (1-35, 5.30)	1.05 (1-4, 0.25)	5192	1950	217
E2E	51426	990	20.34 (1-71, 6.93)	1.56 (1-6, 0.71)	51426	6039	120
SRC	1760	99	39.75 (10-160, 25.50)	4.27 (1-25, 3.33)	1760	77	52
Sem. ESRC	1344	309	24.94 (6-88, 9.52)	2.00 (1-7, 0.90)	2284	859	81
Full ESRC	5361	577	24.07 (5-100, 9.03)	1.92 (1-7, 0.89)	10962	1091	96

the relatively stilted way in which price is communicated in the SRC: the average cost of a meal at a restaurant is always described using a genitive determiner phrase modifying the word 'price' and a simple copula (i. e. '*Restaurant*'s price is *X* dollars'). For this one simple property, our corpus includes: 'costs', 'is', 'has food for', 'with a price of', 'is priced at', 'for N dollars you can eat at X', 'expect to pay N dollars', etc. Table 8 compares the vocabulary size to existing resources.

7.4.2.2 *Text differences by condition*

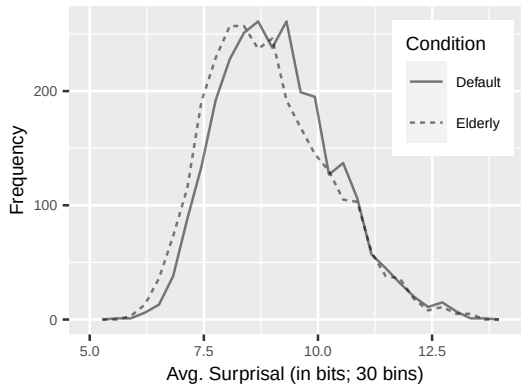


Figure 23: Relative frequency of different average surprisals across the texts in our corpus.

Our data collection included an explicit manipulation of the intended audience to elicit variation with respect to text difficulty. Asking participants to address 'their 85-year-old' grandmother was effective in getting them to produce texts with significantly lower average information density as we hypothesized it would (8.90 vs. 9.11 bits, $p < 10^{-8}$ by Welch's t-test; cf. Figure 23).

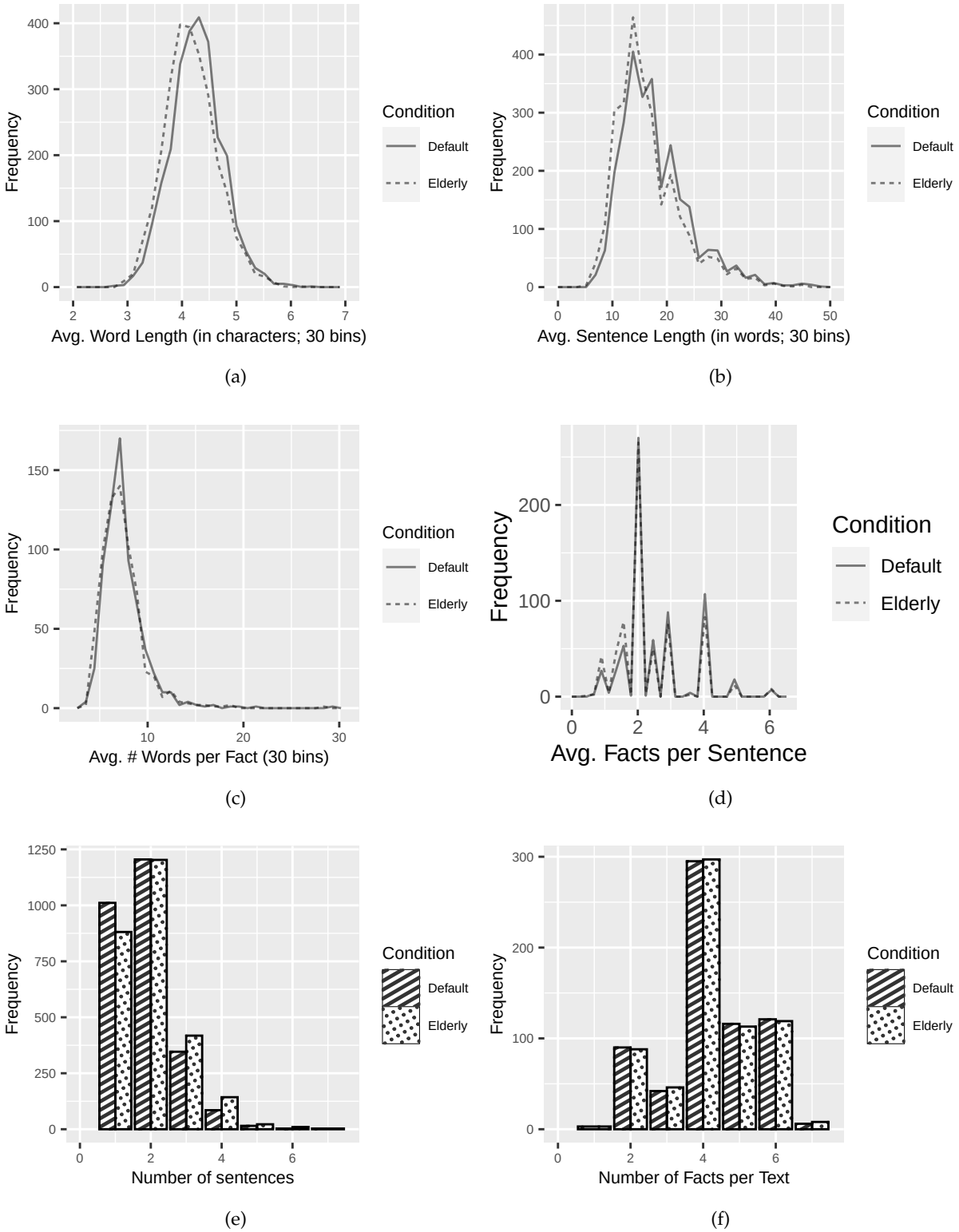


Figure 24: Differences between the DEFAULT and ELDERLY conditions in the ESRC corpus. The means in (c) and (d) are not significantly different; the rest are significant, with p-values specified in Table 9.

Table 9: Properties of the texts in the DEFAULT and ELDERLY conditions, with the significance of the differences between the means based on Welch's *t*-test. *n.s.* = not significant

	Average Value		
text property	DEFAULT	ELDERLY	p
info density	9.11	8.90	$< 10^{-8}$
word length	4.27	4.16	$< 10^{-15}$
sentence length	17.8	16.5	$< 10^{-13}$
# words / REFDA	7.49	7.35	0.2 (n.s.)
# REFDAs / sentence	2.52	2.33	< 0.001
# of sentences / text	1.84	1.98	$< 10^{-8}$
# of REFDAs / text	4.23	4.21	0.8 (n.s.)

Table 9 highlights several other points of comparison between the corpora, with corresponding plots in Figure 24. We observe that the average word length and average sentence length are both shorter for the ELDERLY condition, corresponding to simpler texts according to readability measures like Flesch-Kincaid. Moreover, the average number of facts expressed per sentence is lower and the average number of sentences per text is higher, suggesting that the semantic content is more spread out in the ELDERLY condition. That is, the ELDERLY texts are simpler both in terms of Shannon information density and 'general' information or concept density.

The average number of REFDAs per text and the number of words per REFDA are not significantly different between the conditions, which makes sense because we collected texts for the same sets of REFDAs in each condition.

7.4.2.3 *Meaning representation statistics*

While every text in the corpus is associated with set of propositions and RST relations from the SRC, in this section we focus only on the 1344 texts for which we have manual annotations. Our corpus includes $\approx$ 5700 REFDA tokens consisting of $\approx$ 360 unique REFDA types. Converting these into the CUED dialogue act style, we have 570 unique dialogue acts with 2284 dialogue act tokens. Considering only the combination of slots and not the values for these dialogue acts, there are 107 unique dialogue acts of which 56 occur at least 5 times in the corpus.

At the level of the tree structured text plans, we have 187 unique tree structures when we ignore the labels of the leaf nodes, which are the individual REFDAs. If we collapse all leaf nodes into their parent, we have 102 unique tree structures, of which 34 occur at least five times in the corpus.

7.5 CONCLUSION

We have presented two corpora used in the evaluations reported in this thesis. The SPaRKy Restaurant Corpus is a pre-existing resource, but the Extended SRC is a novel contribution providing a richer resource for learning to generate. Our elicitation interface for corpus collection via paraphrasing will enable other researchers to build similarly varied datasets.

Beyond its utility in the context of this thesis, the variation embedded in the ESRC will be useful to other researchers in the future, as it is the first corpus explicitly designed to include general and elder-directed texts which express the same semantics.

8 EVALUATING NLG SYSTEMS

There is no clear consensus as to how best to evaluate Natural Language Generation (NLG) systems, limiting the ability to compare different systems. However, there is a shared set of terminology and some common norms for both automated and human evaluations.

The next section introduces these norms after which we discuss the factors which have to be considered for human evaluations in particular. We then discuss considerations for system evaluation which do *not* consider the quality of the final texts before highlighting the evaluation methods chosen for this thesis.

These evaluation methods are then used in the Chapter 9 to evaluate the quality of the NLG system presented in this thesis.

8.1 COMMON EVALUATIONS

Given the lack of consensus around automated evaluation metrics for NLG, the gold standard[1] is human evaluations. This section first presents some of the automated metrics (e.g. BLEU) which have been used, along with an explanation of why they are not good enough for the community to agree to use them. We then present the dimensions usually evaluated using human subjects (i.e. fluency and adequacy) before touching on extrinsic evaluations.

8.1.1 *Automated evaluations*

Automated evaluations are appealing during system development as they provide an easy way to tell whether your system has improved after implementing new features or bug fixes. In the ideal case, an automated metric is sufficiently easy to calculate and reliable enough that it can be used as an objective function for machine learning approaches to the task at hand. Unfortunately no existing automated metric can adequately serve these purposes for NLG.

8.1.1.1 *BLEU*

Papineni et al. (2002) introduced the Bilingual Evaluation Understudy (BLEU) score as a diagnostic metric for machine translation. Based

1 The use of this expression appears to be more or less disconnected from the monetary system used for much of the 20th century. In computational linguistics it generally refers to 'the' best standard for evaluation or the best available data for a given task. Sometimes supplementary data created to a lower standard of annotation is called 'silver standard' data by analogy.

on the degree of overlap between n-grams in its translations and n-grams in human-translated reference texts, a system receives a score between 0 and 100. As alluded to in the name[2], this score was never intended to be a substitute for proper evaluation of machine translation systems. Rather, the authors intended BLEU to be used to compare one version of a translation system to later iterations of the same system in order to assess the relative performance of these two incarnations of the system with respect to a particular set of potential translations.

During the first 10 years after BLEU's creation it saw relatively little use in work on natural language generation. Belz & Reiter (2006) investigated the validity of BLEU in the severely constrained domain of weather forecasts. Aside from this, a number of works evaluating OpenCCG's broad-coverage grammar of English by regenerating texts from the Penn Treebank (PTB) used BLEU to see how well the system reproduced the original texts (Espinosa et al., 2010; Espinosa, White & Mehay, 2008; Rajkumar, Espinosa & White, 2011; Rajkumar, White & Espinosa, 2009). In both of these cases the domain of application was sufficiently constrained that BLEU scores were useful for evaluation. Belz & Reiter (2006) conclude that BLEU and similar metrics have potential in applications when the set of possible evaluators is very small and high-quality reference texts are available. However, they also observed that these metrics are biased in favor of NLG systems which select their outputs based primarily on frequency.

In contrast to this work from within the pre-existing NLG community, much recent work on end-to-end or neural NLG emphasizes performance with respect to BLEU scores and related metrics (Dušek & Jurčíček, 2015, 2016; Gu, Liu & Cho, 2019; Karpathy & Fei-Fei, 2015; Kiddon, Zettlemoyer & Choi, 2016; Lebret, Grangier & Auli, 2016; Sha et al., 2018; Shen et al., 2019; Takase et al., 2016; Tran & Nguyen, 2017; Tran, Nguyen & Tojo, 2017; Wiseman, Shieber & Rush, 2018, *inter alia*). Indeed, Gkatzia & Mahamood (2015) found that BLEU-like automated metrics were dramatically more common in non-NLG-specific venues compared to NLG-specific venues and becoming more common in general.

Partly in response to this trend, Reiter (2018) conducted a structured survey of BLEU as it has been used in the NLG and Machine Translation (MT) communities to explore its validity. In examining 284 correlations with human judgements across 34 papers, Reiter found that BLEU serves its intended purpose for MT but does not correlate well with human judgements for other kinds of NLG. He carefully concludes that "the evidence does *not* support using BLEU to evaluate other types of NLP systems (outwith MT), and it does *not* support using BLEU to evaluate individual texts instead of NLP systems" (emphasis in the original), and presents several brief arguments against

2 Being an 'understudy' implies that something is a (usually less-skilled) stand-in or back-up for something else.

BLEU as a primary tool for evaluation in NLG. In particular, the correlations with human judgements appear to be highly context dependent, and BLEU has technological biases that we do not yet fully understand. This means that we simply cannot draw meaningful conclusions about system quality from comparisons of BLEU scores.

Reiter does, however, concede that BLEU can serve as a useful diagnostic for use internally during development. In Section 8.4.2.1 we explore one such diagnostic use: determining how different two systems outputs are from each other.

8.1.1.2 *Other noteworthy automated evaluations*

In discussing BLEU we mentioned that other similar text comparison measures are used to assess text quality relative to some set of reference texts. NIST[3], for example, is an adaptation of BLEU which gives more weight to infrequent *n*-grams (Doddington, 2002). TER (Snover et al., 2005) looks at the minimum number of edits necessary to tranform a text to match some reference, METEOR (Banerjee & Lavie, 2005) targets recall more than BLEU does, and ROUGE (Lin & Hovy, 2003) further shifts the balance toward recall. These other measures are also typically tailored for particular tasks rather than NLG in general. For example, ROUGE is designed to assess summarization systems and SARI (Xu et al., 2016) to assess text simplification systems. There is not strong evidence one way or the other that these metrics are useful for evaluating NLG systems in general.

8.1.2 *Human evaluations*

The point of generating text, spoken or otherwise, is to convey information to human users. Evaluating the most important qualities of a system with human participants therefore remains the gold standard in evaluating NLG systems.

As we shall see, while there is some consensus as to what areas of text quality should be assessed in general, the community has not settled on a single way of asking these questions, reducing the comparability of the results reported by different research teams.

8.1.2.1 *Fluency, Grammaticality, and Readability*

A variety of different questions have been asked under the umbrella of *fluency* or *grammaticality*, usually focusing on the well-formedness of the text or, as Gatt & Krahmer (2018) put it, 'the linguistic quality of the text'. These questions are often designed to assess text quality beyond the syntactic or sentential level, often considering the pragmatic and semantic felicity of a text as well. Indeed, these questions

3 Named for the National Institute of Standards and Technology in the USA

are often framed in terms of *readability* as well, opening the door to assessments of *clarity* and *understandability*.

Table 10 lists a number of questions or prompts along with the scales used for assessing texts for these criteria. Note that a number of the scales are either underspecified (i.e. the original papers presenting the findings do not indicate how the question posed to subjects was formulated) or rely on the subjects' understanding of a particular term, like 'fluency' or 'grammaticality'. Those systems which provide a more explanatory framing of the question or tie the different scores to a particular description help to ensure that subjects are rating the texts according to the same properties that the researchers want to examine.

The ultimate goal of a written text is to convey some meaning to a reader, which requires that the text is 'readable', as highlighted by several of the evaluation questions in Table 10. However, as we have previously remarked (Howcroft & Demberg, 2017), this is a precondition for comprehension rather than a measure of understandability. Therefore we include in Table 11 a sample of some of the questions previous researchers have used to assess the 'understandability' or 'clarity' of their system's texts.

Ideally future methodological work will establish reliable question and scale formulations for assessing these issues and provide evidence of their validity. In the meantime, we use our own formulation to assess fluency and grammaticality on a sliding scale, futher detailed in Section 8.4.2.2.

8.1.2.2 *Adequacy, Completeness, and Informativeness*

The other major dimension of text quality which is typically evaluated has to do with the semantic content of the text, focusing on the "accuracy, adequacy, relevance or correctness relative to the input, reflecting the system's rendition of the content" (Gatt & Krahmer, 2018). While the typical approach provides subjects with some representation of the non-linguistic input to the system and asks them to assess the completeness of the text with respect to the information the system aims to convey, researchers have again used a wide variety of different framings for asking these questions.

Table 12 provides an overview of approaches since 2002. Here we see that the questions and prompts vary with respect to their emphasis (i.e. the informativeness versus the adequacy or completeness of a text).

They also differ with respect to what kind of reference they provide to subjects for evaluating completeness. For example, Callaway & Lester (2002) ask general questions about whether a text is logical and informative without providing any reference, while Stent, Prasad & Walker (2004) use reference texts and Mitchell et al. (2012) offers the images which their systems are designed to describe as a reference.

Table 10: Sample of published approaches to eliciting human judgements of grammaticality, fluency, or readability.

Source	Question or Prompt	Scale	Category
Callaway & Lester (2002)	"Grammaticality: How would you grade the syntactic quality of the story?" "Flow: How well did the sentences flow from one to the next?" "Readability: How hard was it to read the prose?"	5 point likert scale presented as US grade scale (A, B, C, D, F)	GRAMMATICALITY FLUENCY READABILITY
Stent, Prasad & Walker (2004)	"How do you judge the fluency of Sentence B?"	"It is (flawless | good | adequate | poor | incomprehensible)"	FLUENCY
Espinosa et al. (2010)	*Question not given*	5 = perfectly grammatical 4 = awkward or non-native; punctuation errors 3 = agreement errors or minor syntactic problems 2 = major syntactic problems, such as missing words 1 = completely ungrammatical	FLUENCY
Angeli, Liang & Klein (2010)	*Question not given*	5 = flawless 4 = good 3 = non-native 2 = disfluent 1 = gibberish	FLUENCY
Belz et al. (2011)	"Your task is to decide how well the highlighted sentence reads; is it good fluent English, or does it have grammatical errors, awkard constructions, etc."	Slider from 0 (couldn't read worse; frownie) to 100 (couldn't read better; smiley)	GRAMMATICALITY, FLUENCY
Mitchell et al. (2012)	"This description is **grammatically correct**."	5 point Likert scale of agreement	GRAMMATICALITY
Kuznetsova et al. (2012)	*Question not given*	5 = perfect 4 = almost perfect 3 = 70-80% good 2 = 50-70% good 1 = totally bad	GRAMMATICALITY
Elliott & Keller (2014)	"Grammaticality: give high scores if the description is correct English and doesn't contain any grammatical mistakes"	5 point Likert scale	GRAMMATICALITY
Gyawali & Gardent (2014)	*Question not given*; may have used "Is the text easy to read?" for fluency	Slider from -50 to 50	GRAMMATICALITY, FLUENCY
Novikova, Dušek & Rieser (2017)	"How do you judge the overall quality of the utterance in terms of its grammatical correctness and fluency?"	6-point likert scale	GRAMMATICALITY, FLUENCY

Table 11: Sample of published approaches to eliciting human judgements of understandability or clarity.

Source	Question or Prompt	Scale
Bangalore, Rambow & Whittaker (2000)	"How easy is this sentence to understand?"	7 = extremely easy 4 = just barely possible 1 = impossible
Oh & Rudnicky (2002)	"which system's responses were easier to understand?"	Forced choice
Belz et al. (2011)	"How clear (easy to understand) is the highlighted sentence within the context of the text extract?"	Slider from 0 (couldn't read worse; frownie) to 100 (couldn't read better; smiley)
Hunter et al. (2012)	"The BT-Nurse summary was easy to understand."	'indicate agreement, disagreement, or neutrality'

Because there remains no standard representation of input for NLG systems and they are often applied for different tasks, the diversity of evaluation methods for this dimension is more defensible. For our own work we have chosen to focus on evaluating the presence or absence of key facts (i.e. slot-value pairs), the insertion of unnecessary or incorrect content, and the expression of two discourse relations, as detailed in Section 8.4.2.2.

8.1.2.3 *Other noteworthy human evaluations*

In addition to these two aspects of text quality, researchers often ask subjects to rate the overall quality of a text, allowing users to define quality however they see fit.It is also common to ask subjects to rate a text for naturalness or humanlikeness (Cahill, 2009; Howcroft, Nakatsu & White, 2013; Mitchell et al., 2012; Novikova, Dušek & Rieser, 2017, 2018, i.a.), helpfulness or usefulness (Hunter et al., 2012, i.a.), and ordering of information (Callaway & Lester, 2002; Mitchell et al., 2012, i.a.). Of course, other aspects like genre fit (Callaway & Lester, 2002), perceived personality (Mairesse & Walker, 2011; Mairesse & Walker, 2007; Oraby et al., 2018), and style judgements (Lester & Porter, 1997) may be relevant depending on the task.

One interesting approach which has not been revisited recently for evaluation is the idea of using post-edit information in order to assess the quality of a system's output (Sripada, Reiter & Hawizy, 2005). Post-editing has been used successfully in MT evaluations and leverages human intelligence to gauge how far off a system's output is from a correct realization. This approach has the added benefit of

Table 12: Sample of published approaches to eliciting human judgements of adequacy/completeness and informativeness.

Source	Question or Prompt	Scale	Category
Callaway & Lester (2002)	"Logicality: Did the story omit crucial information or seem out of order?" "Detail: Did it have the right amount of detail, or too much or too little?"	5 point likert scale presented as US grade scale (A, B, C, D, F)	ADEQUACY, INFORMATIVENESS INFORMATIVENESS
Oh & Rudnicky (2002)	"which system offered you more information?"	Forced choice	INFORMATIVENESS
Stent, Prasad & Walker (2004)	"How much of the meaning expressed in Sentence A is also expressed in Sentence B?"	(all | most | half | some | none)	ADEQUACY
Angeli, Liang & Klein (2010)	*Question not given*	5 = perfect 4 = near perfect 3 = minor errors 2 = major errors 1 = completely wrong	ADEQUACY
Espinosa et al. (2010)	*Question not given*	5 = All the meaning of the reference 4 = Most of the meaning 3 = Much of the meaning 2 = Meaning substantially different 1 = Meaning completely different	ADEQUACY
Mitchell et al. (2012)	"This description **describes the main aspects** of this image." "This description **does not include extraneous** or incorrect information."	5 point Likert scale	ADEQUACY ADEQUACY
Hunter et al. (2012)	"The BT-Nurse summary is accurate"	"indicate agreement, disagreement, or neutrality"	ADEQUACY
Kuznetsova et al. (2012)	*Question not given*; "cognitive correctness" and "relevance"	5 = perfect 4 = almost perfect 3 = 70-80% good 2 = 50-70% good 1 = totally bad	INFORMATIVENESS
Elliott & Keller (2014)	"Action: give high scores if the description correctly describes what people are doing in the image" "Scene: give high scores if the description correctly describes the rest of the image (background, other objects, etc)"	5 point Likert scale	ADEQUACY ADEQUACY
Gyawali & Gardent (2014)	*Question not given*; may have used "Does the meaning conveyed by the generated sentence correspond to the meaning conveyed by the reference sentence?" for fluency	Slider from -50 to 50	ADEQUACY
Novikova, Dušek & Rieser (2017, 2018)	"Does the utterance provide all the useful information from the meaning representation?"	6-point likert scale	ADEQUACY

producing new reference texts which can be used for training the system further.

8.1.3 *Extrinsic evaluations*

intrinsic

extrinsic

The kinds of human evaluations we have discussed so far are *intrinsic*: they evaluate the quality of a resulting text primarily as a text. Text is usually produced to serve a particular purpose, however, so it is worth asking: does the generated text help achieve the desired outcome? And, if so, does it perform better than a reasonable alternative? Such evaluations of text generation *in situ* as part of a larger system are called *extrinsic* evaluations.

The TIPSTER SUMMAC Text Summarization Evaluation (Mani et al., 1999) focused on only one kind of NLG: summarization. Here they used two extrinsic tasks based on common tasks faced by government analysts: determining source document relevance and categorizing a document based on a summary.

Around the same time, Young (1999) explored a planning-based generation system for producing instructions (plans of action) to assist users in accomplishing a task such as checking out a book from the library or sending an email using a particular piece of software. Young used an empirical evaluation judging how long it took subjects to complete the task and found that the instructions produced by their system helped subjects complete the task more quickly.

More expansive modes of extrinsic evaluation are also possible, although generally uncommon. The STOP program (Reiter, Robertson & Osman, 2003) aimed to help smokers quit smoking, but found no evidence that the texts generated specifically for a particular patient helped above and beyond a simple flyer containing general information about smoking cessation. Partnering with medical practitioners to conduct a thorough evaluation, however, took more than a year and a half and cost 75 thousand GBP. Similarly, the BabyTalk project (Gatt et al., 2009; Portet et al., 2009) included a number of extrinsic evaluations with varying degrees of ecological validity, ultimately costing about 20 thousand GBP and six months to evaluate.

Perhaps these costs are the reason that relatively few published studies conduct a full extrinsic evaluation. Gkatzia & Mahamood (2015), for example, found that only 15% of papers published in the previous decade had included some form of extrinsic evaluation. In response to this observation Hastie et al. (2016) published their own experience with the extrinsic evaluation of the NLG component of a telephonic restaurant recommendation system. Their findings showed that there was no objective difference in task success between the two systems under consideration, but that users perceieved themselves as having significantly higher task success with one system compared to the other. This reinforces the idea that understanding

the full subjective impact of a system on users satisfaction with a system requires an extrinsic evaluation.

Such subjective findings can also be quite surprising. Kousidis et al. (2014) developed an incremental dialogue system which could interrupt itself based on situational cues and resume speaking when it was safe to do so. Their extrinsic evaluation revealed that such an adaptive system resulted in safer driving but was dispreferred by users!

8.2 DESIGNING HUMAN EVALUATIONS

How best to evaluate an NLG system depends in part on the questions we want to answer. There are several dimensions along which our approach can vary, which we survey here.

8.2.1 *Number of Systems Presented*

Presenting only one system precludes the ability to make explicit comparisons, but this can be a good thing. When a system is deployed, users will have different expectations regarding the capability of such systems in general.

If we want to assess how users will react to a new system upon release, it is helpful to know what they think of that system on its own. Introducing texts from another system with different capabilities (e.g. a system which uses colloquialisms and salutations to build a sense of familiarity with the user) ensures that users will consider those differences in their evaluation. If, however, users would not have considered this possibility otherwise, we may find that users 'disprefer' a system, despite the fact that they would appreciate it on its own merits taken alone.

Of course, we often want to explicitly compare different NLG systems, in which case it may make more sense to run an explicitly comparative evaluation rather than having subjects score texts separately and collecting only implicit comparisons. For discussion of this issue, see the section on scoring vs. ranking (Sec. 8.2.4).

8.2.2 *Number of texts presented*

The more texts each subject sees, the greater the chance that they will identify the particular differences we are evaluating. This is a common problem in experimental design, usually addressed by adding 'filler items' so that subjects remain blind to the goal of the study. In the context of evaluation for NLG, this can manifest in terms of subjects spotting keywords that occur in the texts that they usually prefer. When subjects have to evaluate many texts and payment is fixed, the incentive is high to select a few easily identifiable features in this way to speed through the task.On the other hand, having more ratings

from each subject makes it easier to estimate by-subjects variation and assess what quirks are unique to each subject.

8.2.3 *Simultaneous text presentation*

When presenting texts to human subjects for evaluation, researchers can choose to present the texts one after another or to present two or more texts at the same time.

For example, Howcroft, Nakatsu & White (2013) provided each of their human evaluators with all of the texts-to-be-evaluated at the same time, despite not framing their evaluation as an explicitly comparative task. This allowed subjects to revisit their evaluations if, for example, they gave one text the worst score and later encountered a text which they found worse, forcing them to shift their use of the scale.

forced choice

More typical is the presentation of only two texts at a time in a *forced choice* task where subjects choose which text is better along some dimension. This forced choice format for text comparison makes evident all details in which a pair of texts differ. This is quite similar to ranking multiple texts, which we discuss in the next section.

8.2.4 *Scoring vs. ranking*

Having subjects score texts on a continuous or discrete scale provides a means of assessing the overall quality of a system independently. These scores can be used to provide an implicit ranking, although such rankings are probably less valid when they are collected between-subjects, unless sample sizes are substantially larger than typical for NLG evaluation.

Ranking texts requires direct comparison and has the benefit of aligning with the frequent objective of NLG evaluations: explicit comparison of different NLG systems. In the case where subjects are 'ranking' two texts, this reduces to a forced choice task.

One problem with rank-based approaches is that the number of pair-wise comparisons necessary for a complete evaluation grows exponentially with the number of systems to be evaluated, although some have proposed approaches like the TrueSkill ranking algorithm to address these problems (Novikova, Dušek & Rieser, 2018).

8.2.5 *Blinding with respect to system identity*

Usually in experimental design we want subjects to be blind to the different experimental conditions. If, however, we want to get an overall impression from users regarding their preferences for one system over another, it may be worthwhile to label the different systems with simple names (e.g. System A and System B) and ask subjects to rate

the quality of the texts produced by or their interactions with the different systems. This line of evaluation provides subjects an opportunity to form a more holistic impression of what each system has to offer.

8.2.6 *Single vs. Multiple Questions/Dimensions of evaluation*

The standard practice is to evaluate NLG systems with respect to both adequacy and fluency, usually having the same subjects score each text along both dimensions. However, Novikova, Dušek & Rieser (2018) found that scores for adequacy and fluency were often correlated when the same subject rated both for the same texts.

While decomposing the evaluation into a number of separate surveys, each focusing on a single question, can streamline the experience for crowdsourced participants, this approach is not without drawbacks, since there is overhead involved with each individual experiment. Moreover, for languages without large populations of crowdsourcing workers it is often sufficiently difficult to recruit subjects that it is important to collect as many judgements from each subject as possible.

8.2.7 *Lab- vs. web-based evaluation*

In the last ten years it has become extremely common to deploy NLG evaluations online, typically using crowdsourcing platforms. This makes it considerably easier to conduct evaluations of text quality for a large sample of output texts for each system with a large population of less WEIRD[4] participants.

During the same period, web browsers and high-bandwidth internet access have become more reliable, making it easy to deploy these experiments as well. However, it is still not unusual to run into unusual hardware or software problems when deploying crowdsourcing experiments, especially if the goal is to evaluate the system in a spoken dialogue context. Therefore we must always run thorough tests to ensure that any web-based evaluation will function as intended.

8.2.8 *Demographic and other information collected*

While demographics are not usually central to an NLG evaluation, it is important to characterize the subject population in order to under-

4 This acronym stands for "White, Educated, Industrialized, Rich, Democratic" and acknowledges the fact that the population of convenience used for most published research typically consists of undergraduates at univiersities in Europe and North America. Not only are these subjects not representative of the populations of the countries that they live in, but they are wildly unrepresentative of humanity as a whole.

stand how well the results of any human evaluation may generalize to other populations of users. Typical survey demographics include age and gender identity, but the most important demographics for us usually relate to language background. In particular, we usually want to collect judgements from native speakers of the target language, although it is reasonable to collect some amount of judgements from non-native speakers if they are part of the target user population.

Also important for our purposes is the extent to which users are already familiar with computer-generated language and (digital) personal assistants.[5] When subjects are used to interacting with Google Assistant, Siri, Alexa, Cortana, or other digital assistants, they develop expectations about the kind of language that computers are able to produce which can shape the results of our evaluation.

8.3 EVALUATING SYSTEMS (NOT TEXTS)

While human evaluations are usually taken as the gold standard for assessing the quality of an NLG system, the overall text quality needs to be weighed against other properties of the system. Practical limits including the amount of training data needed and the computational resources required can have a big impact in the decision of what model should be deployed in a particular situation.

In addition to these considerations, it is also possible to analyze the capabilities and accuracy of a system from a formal perspective, determining in principle what it is capable of.

8.3.1 *Practical limits*

While we are interested in NLG as a research community, the goal of most NLG systems is to serve a practical purpose in some application. To this end it is worth considering some of the practical aspects of deploying a system.

When deploying a system it is important to consider the time required to generate a response, as well as what kinds of resources are required to produce that text in the first place. Any pre- or post-processing required to achieve good text quality can further introduce delays and new dependencies to the project. Finally, since we are interested in machine learning-based approaches to NLG, we must consider the amount of training data required to achieve reasonable performance.

5 Not to be confused with Personal Digital Assistants, which were briefly popular in the early 2000s.

8.3.1.1 *Run-time and memory/disk requirements*

For offline text-generation, run-time characteristics are less relevant, but in the context of dialogue systems it is extremely important to be able to generate responses quickly[6]. Although some NLG systems have reported runtimes for individual texts (e.g., Gabriel, 1988; Knight & Hatzivassiloglou, 1995) or for some test sets (e.g., Becker, 1998; Belz, 2005, 2008; Carroll et al., 1998; Koller & Hoffmann, 2010; Koller & Petrick, 2011; Lavoie & Rambow, 1997; Schwenger et al., 2016), this is by no means the norm in the field.

Even more rare is any discussion of the amount of memory or disk space or processing power necessary to run an NLG system. For individual papers or projects this makes sense, as we are usually evaluating the feasibility of a particular approach to NLG and assessing its theoretical strengths and weaknesses. However, this makes our literature less useful to practitioners who want to assess whether or not a particular approach makes sense for their use case, as the runtime constraints on an offline report generator are quite different from a personal assistant deployed as a dialogue system on a mobile device with a faulty internet connection, for example.

8.3.1.2 *Pre- & Post-processing requirements*

Deploying NLG systems requires providing them with an appropriate input representation and presenting users with an acceptable textual or spoken output, but most of our research focuses on just one part of this process. Even so-called end-to-end systems require basic NLP preprocessing to normalize, tokenize, and perhaps delexicalize inputs for training and require complementary post-processes. These processes are common plumbing tasks in NLP: we know that we need normalization, etc, but we are more interested in how our models treat the normalized data than the process of normalizing it. Such decisions, however, can have dramatic impacts on system performance and need to be properly documented for replication.

In addition to obvious sorts of pre- and post-processing, the line can become blurred between the NLG system under consideration and other related tasks. For example, is the re-ranking stage of an overgenerate-and-rank-style system a part of the generation process, or a post-process? We might, for example, seek to minimize the amount of reranking that is necessary by improving the search characteristics of our system so that we are less prone to *over*generate. In other instances, however, we may prefer to consider the generation system as the sum of all pre- and post-processing necessary for deployment.

6 See, for example, White (2004)'s discussion of the efficiency of OpenCCG as deployed in several dialogue systems.

8.3.1.3 *Amount of training data required*

In developing Machine Learning (ML)-based systems for NLG we must consider the amount of training data available for our models to learn from. In MT, for example, it is relatively feasible to collect large parallel corpora for dominant languages, allowing the use of data-hungry ML methods. For generation, however, we are unlikely to have access to semantically annotated corpora consisting of hundreds of thousands of texts, let alone millions or billions of training items. Therefore it is important to be able to achieve good quality generation from hundreds or thousands of training items.

8.3.2 *Well-formedness of the resulting rules*

For end-to-end NLG systems the only possible intrinsic evaluations are those evaluating the quality of the texts. However, when developing a rule-based approach, we can also evaluate the quality of the individual rules created by our developers or learnt by our algorithms.

Consider the work of White & Howcroft (2015), who used a template-based system for learning clause-combining operations over logical forms. To evaluate their system's performance and the impact of additional training data thereon, they manually inspected each of the top twenty rules learnt by the system after exposure to 20, 40, 60, ..., up to 200 training pairs. Categorizing rules as 'good' (roughly: a rule that could have been written by a grammar engineer), 'overspecified' (containing more lexical specification than necessary but otherwise accurate), and 'bad' (containing invalid alignments or other problems), they found that more than three quarters of the top twenty rules were acceptable ('good' or 'overspecified') and that overall the system learned roughly equal proportions of good, overspecified, and bad rules. This evaluation led them to conclude that (1) keeping the highly ranked rules provides a reasonable heuristic for rule quality but also that (2) this approach would work best as a development tool to aid grammar engineers rather than as a fully independent system.

In the present work we take a similar approach to evaluating the quality of the rules learnt by our system. In addition to providing an alternate means of assessing the quality of a system, this evaluation can contribute to an error analysis of the kinds of errors we observe later in the resulting texts.

8.4 EVALUATION METHODS USED IN THIS THESIS

In the following chapters we will focus on a small set of automated evaluations for examining differences between different versions of the same model. For evaluating text quality we use two methods of

human evaluation, one quick and efficient to be applied to a large number of models by a single user and a more extensive evaluation to be deployed via crowdsourcing.

8.4.1 *Automated evaluations*

The most basic question we need to answer is this: can the rules our system learns be used to generate new texts for unseen text plans? The simplest metric for answering this question is to apply the rules to a held out set of text plans and see what proportion of the text plans we are able to (1) generate logical forms for and (2) generate full texts for. This gets to the most basic question at the heart of any machine learning evaluation: does our model generalize to unseen data or have we overfit our model to the training data?

This raises the issue of trade-offs with respect to generalizability & quality and consideration of the interaction between rule & text quality. For example, we may accept reduced generalizability in exchange for higher text quality for those cases where we are able to generate a text. This is analogous to the more general precision-recall trade-off.

Once we have two models which can generate text, we need to know how different the texts produced by these two models are. Indeed, it is possible for one model to have less coverage than another, but to produce exactly the same texts for the subset of text plans that it works on. We could give all of these texts to human evaluators, but if there is no substantial difference between the outputs of two models it does not make sense to compare them.

To determine how different the outputs of two models are, we use the BLEU score as a similarity metric. Importantly, we are not using BLEU to evaluate the quality of a generated text in comparison to some reference text, but rather we are simply using it to compare the output of one model with the output of another. When two models differ substantially, the BLEU score for these texts will be lower, and we can then investigate the texts more fully in a human evaluation.

8.4.2 *Assessing text quality*

We conduct human evaluations of text quality in two settings. First, we use a quick-and-easy set up for evaluating the impact of incremental changes. Second, we deploy a crowd-sourced evaluation to evaluate the 'final' systems of interest.

8.4.2.1 *Incremental Comparisons*

During development we do not want to deploy a full crowdsourcing experiment to assess every incremental change. The purpose of these evaluations is to collect rapid judgements: are the texts produced by

one initialization, set of parameters, or other system variation better than the texts produced by another?

For these purposes we use a simple Python script which displays the text plan input to our sentence planner along with two texts. The order of the texts is randomized, so that the scorer does not know which system is which. The scorer then says whether the first text is better than the second text, the texts are roughly equal, or both texts are too bad to be worth evaluating. For scorers familiar with the format it is quick and easy to assess all of the texts in the dev set, comparing about 100 TP-text pairs in about 20 minutes.

The outcome of this evaluation is a quick sense of whether one system produces substantially better texts than the other, and therefore this approach guides us towards better configurations during development. Moreover, if two systems perform comparably, with each surpassing the other in a similar number of cases, then we know that these two systems are also worthy of further exploration, either in terms of a full human evaluation or an error analysis to understand their relative strengths and weaknesses, or both.

8.4.2.2 *Crowdsourced Human Evaluation*

As in our corpus elicitation experiments (Sec. 7.4.1.1), we used LingoTurk (Pusse, Sayeed & Demberg, 2016) for these experiments. We began by designing a new evaluation interface suited to the goals of our evaluation, pictured in Figures 25 through 28.

After reading the experiment description and working through two example texts, participants are presented with a series of questions about a single text in context. Based on the original task goals from the SPaRKy Restaurant Corpus (cf. 7.3.1), we generate a fake user query based on a template to serve as context for the system-generated utterance.

After reading the text the participant scores the grammaticality text on a sliding scale designed to provide guidelines for different relative levels of grammaticality while allowing for fine-grained differentiation on the part of the subject. The scale can be seen in Figures 26 through 27, with each of these four levels pinned to the values 0, 33, 67, and 100 on the sliding scale. Unlike traditional Likert-scale based evaluations, this provides categorical guidance while allowing for more fine-grained distinctions. Previous research also suggests that raters tend to prefer sliders to Likert scales (Belz & Kow, 2011).

Once this assessment of fluency is complete, subjects proceed to rate the semantic completeness of the generated text. Each of the individual 'facts' intended to be expressed by the system is presented in a table where participants can click to mark if item as 'missing'. Participants also note if the system has added any unintended details to the text. This is important because our system can extract rules which

About this experiment

Study Title: Evaluating Text Generation (April 2019)
Researchers: David M. Howcroft, Vera Demberg, Dietrich Klakow
Sponsor: SFB 1102, Saarland University

Please read the following information carefully.

Purpose
The purpose of this study is to compare the output of systems for generating text in English. This will help us to understand which methods are most appropriate for developing these systems.

Procedures
You will read 20 short texts one at a time. After each text you will be asked to rate the correctness of the English in the text. Then you will use a checklist to mark any facts which do not appear in the text and to indicate what order the facts appeared in. After this we ask you to tell us if the text contains any extra information not in the checklist. Finally, you will provide one suggestion for how the text could be improved.

The entire experiment should take about half an hour.

Risks and Benefits
Risks to participants in this study are considered minimal. You will receive payment for your participation in the survey, but there will be no other benefit to you. **Note:** we use screening questions in the survey to which we already know the correct answer to make sure that you are paying attention. If you get more than 25% of these questions wrong, we will discard your data and you will not be paid.

You can withdraw from the study at any time simply by stopping the experiment.

Confidentiality
Responses to this survey will be published along with our analysis, but no identifying information will be published. In particular, your Participant ID will not be published.

Contacts and Questions
If you have questions about this study or would like to be informed about our findings, please contact David M. Howcroft at howcroft@coli.uni-saarland.de.

Next

Figure 25: The consent form used for one of our evaluations.

erroneously include other content and the neural network models we are evaluating as baselines often 'hallucinate' extra material as well.

Recall that the purpose of learning sentence planning rules to map discourse structure to dependency trees for realization is to ensure a richer expression of the intended semantics. However, we do not want to have to instruct participants in how to interpret a text plan in order to assess how much a text matches it. Therefore we address the higher level semantic structure of the text in two ways. First, while completing the simple semantic adequacy task, participants are asked to drag-and-drop the facts into the same order that they appear in in the system's response to the user. This provides our first insight into the level of semantic control present in the system, measuring the extent to which it learns to express facts in the same order that they are presented in the text plan.

The second question focuses on the discourse relations our system learns to express. Depending on the text plan, subjects are asked to assess whether or not a text expresses a contrast relation or a justification relation, as shown in Figures 26 through 27.

Finally, the evaluation for each text concludes with an open-ended question for the subject: we ask for a single suggestion for improving the texts. This question serves three purposes: (1) it aims to encourage the subjects to pay attention to the texts they are scoring, (2) it pro-

Instructions

In this study, we are looking at the quality of system responses for a dialogue system.

On each slide you will see a prompt from a user and one possible system response, as shown below.

Please read the following dialogue, focusing on the system response:

User: Tell me about these West Village restaurants.
System: Da Andrea has good decor and Da Andrea has very good service, with good food quality. Gene's has decent decor.

Following the dialogue you will be asked to rate how correct the English is in the system response. In this example, the text is clearly understandable and the English is grammatically correct, but it's also a bit clumsy. Please rate the text accordingly on the following scale.

Is the English in the system response correct?

This makes no sense! Its "word salad"	Major problems; very difficult to understand	Minor problems or awkward phrasing; mostly understandable.	Perfectly correct English

After rating the correctness of the system response, you will check if the system forgot to mention anything. You will see a table of information which is supposed to be included by the system and check a box to mark if any of the information is missing. One fact is completely missing in this example. Please respond accordingly.

Notice that each row has a double-sided arrow at the beginning. This means that you can drag and drop the facts to re-organize them. For the remaining facts, all the facts which the system actually expressed, you need to drag and drop them until they are in the correct order. Give that a try here.

Which (if any) of the following facts are missing from the system response?
Please sort the remaining facts so they match their order in the text.

		Missing
(Da Andrea) decor	good	X
(Da Andrea) service	very good	X
(Gene's) decor	decent	X
(Gene's) service	good	X

After the table we will repeat the text for you so you don't have to scroll too much to answer the next question.The next question is about whether any information was *added* to the text. There is one extra fact in this text. Check "Yes" to enable the text box where you can write down what they are.

Here is the dialogue again, repeated for your convenience.

User: Tell me about restaurants in the west Village.
System: *Da Andrea has good decor and Da Andrea has very good service, with good food quality. Gene's has decent decor.*

Does the system response include any extra details?

Yes No

For some texts you will be asked about the high-level structure of the text. In this example, we are being asked about whether the text makes a comparison. This text is a bit ambiguous, because it tells us about two different restaurants. Maybe it is doing that in order to implicitly compare them. Read the text and decide for yourself what you think the best answer is for this example.

Does the following text compare or contrast (at least) two things?

Yes kind of No

Finally there's a chance to be creative! The last task for each dialogue is a simple one: please give us one suggestion for how to improve the existing system response. For example, this one repeats "Da Andrea" unnecessarily, so you might suggest using "it" or saying "Da Andrea has good decor and very good service". Feel free to also make stylistic suggestions, like "start by telling me how many restaurants there are" or "food quality is a weird phrase; just say the food is good!"

That's it! You'll see 20 texts and answer those questions for each of them

Before you get started, let's look at one more example...

Next

Figure 26: Instructions given to subjects (part 1).

Instructions

Please read the following dialogue, focusing on the system response:

User: Recommend a cheap restaurant.
System: *Amy's Bread with cafe restaurant in Midtown. has best the overall quality selected among, with food food quality and decent service. Cost cost dollars.*

The text in this example has many more problems. The first 'sentence' probably means that Amy's Bread is a cafe in Midtown, but it's not grammatically correct and we have to work to interpret it. Similarly, the rest of the text has a variety of errors in word order, repeated, and missing words. This text is pretty bad (almost word salad!) but still partially understandable.

Is the English in the system response correct?

This makes no sense! Its "word salad"	Major problems; very difficult to understand	Minor problems or awkward phrasing; mostly understandable.	Perfectly correct English

Which (if any) of the following facts are missing from the system response?
Please sort the remaining facts so they match their order in the text.

		Missing
(Amy's Bread) quality	best	X
(Amy's Bread) price	12	X
(Amy's Bread) neighborhood	Midtown West	X
(Amy's Bread) cuisine	Cafes	X
(Amy's Bread) food quality	excellent	X
(Amy's Bread) service	decent	X

Here is the dialogue again, repeated for your convenience.

User: Recommend a cheap restaurant.
System: *Amy's Bread with cafe restaurant in Midtown. has best the overall quality selected among, with food food quality and decent service. Cost cost dollars.*

Does the system response include any extra details?

◉ Yes ○ No

Some of our texts are meant to *justify* a claim. This text was supposed to explain *why* "Amy's Bread" is the best restaurant, but it doesn't really try to justify this claim, so we should answer "no" to the next question. Other versions of this text might express the justification explicitly, saying "Amy's Bread is the best restaurant, because it has good food quality..." or "Since it has good food quality and is in Midtown, Amy's Bread has the best overall quality." When you're answering this question, just ask yourself, "Did the system give a reason for one of its claims?"

Does the following text make a claim and provide supporting facts for that claim?

○ Yes ◉ No

This text invites a lot of suggestions for improvement. For example, "sentences need to have subjects" or "sentences need to have verbs" or "don't repeat the same word multiple times in a row." What suggestions can you think of?

Next

Figure 27: Instructions given to subjects (part 2).

Please follow the instructions.

Question 1 of 20

Please read the following dialogue, focusing on the system response:

User: Recommend a French restaurant.
System: *Le Marais has the best overall quality among the selected restaurants. It has decent decor. This Chinese, Latin American restaurant has very good food quality. Its price is 44 dollars. It has decent decor. It has the best overall quality among the selected restaurants.*

Is the English in this text correct?

This makes no sense! Its "word salad"	Major problems; very difficult to understand	Minor problems or awkward phrasing; mostly understandable.	Perfectly correct English

Which (if any) of the following facts are missing from the system response?
Please sort the remaining facts so they match their order in the text.

		Missing
(Le Marais) quality	best	X
(Le Marais) service	decent	X
(Le Marais) price	44	X
(Le Marais) decor	decent	X
(Le Marais) cuisine	French,Kosher,SteakHouse	X
(Le Marais) food quality	very good	X

Here is the dialogue again, repeated for your convenience.

User: Recommend a French restaurant.
System: *Le Marais has the best overall quality among the selected restaurants. It has decent decor. This Chinese, Latin American restaurant has very good food quality. Its price is 44 dollars. It has decent decor. It has the best overall quality among the selected restaurants.*

Does the text include any extra details?

Yes No

Does the following text make a claim and provide supporting facts for that claim?

Yes No

Please suggest one improvement for the text above:

Next

Figure 28: An example evaluation screen.

vides insight into how much effort the subjects are putting into the evaluation, and (3) it provides a set of textual properties that might be worth considering as a source of possible improvements in future research.

8.5 CONCLUSION

In this chapter we have explored current approaches to NLG evaluation, automated & otherwise, and highlighted important considerations in designing human evaluations. We also touched upon issues related to evaluating an NLG system apart from the quality of its outputs before presenting in detail the evaluation methods used in this thesis.

In the next chapter we examine particular models instantiating our framework for learning sentence planning rules and present the results of our system & text evaluations.

Part III

MODELS AND EVALUATIONS

The meat of the thesis, this section describes the models we explored and details their evaluation.

9 INDUCING AND GENERATING FROM A SYNCHRONOUS TREE SUBSTITUTION GRAMMAR

This chapter introduces our statistical model for Synchronous Tree Substitution Grammars (sTSGs) over text plans and logical forms as well as the Gibbs operators used for inference. Using the implementation of our framework described in Section 6.2 we induce several sTSGs and evaluate them following the procedures described in Section 8.4 under several different training+testing conditions.

This chapter represents one of the major contributions of this thesis with respect to Machine Learning (ML) for Natural Language Generation (NLG).

9.1 HIERARCHICAL CRP FOR TSG DERIVATIONS

Before defining the distributions over pairs of Text Plans (TPs) and Logical Forms (LFs) (*TreePairs*), we should define the base distributions over TPs and LFs alone. As in Chapter 5, we present this model in terms of a generative process which begins with sampling an elementary tree for the root of the tree and then repeating this sampling procedure for each frontier node in the expanded tree.

In this model the states ($q \in Q$) associated with frontier nodes for the Tree Substitution Grammar (TSG) are not part of speech or phrasal categories, but rather the states are what we refer to as *tree locations*. For a given node n the location $l(n)$ is the label of its parent node along with the label of the inbound arc from its parent to it.

tree locations: *(node label, arc label) pairs which identify a position in a tree with labeled arcs, potentially non-uniquely*

Since each tree location q corresponding to a frontier node in the expanded tree is completely determined by the current expansion, we only need to define a distribution T over possible elementary trees e conditioned on q:[1]

$$T|q \sim \mathrm{DP}(1.0, P(e|q)) \tag{39}$$

$$\begin{aligned} P(e|q) = & N(n(\mathrm{root}(e))|q) \\ & \Pi_{a \in a(\mathrm{root}(e))} A(a|n(\mathrm{root}(e))) \\ & \Pi_{child \in \mathrm{children}(\mathrm{root}(e))} P(child|q(child)), \end{aligned} \tag{40}$$

where N and A are Dirichlet processes over possible *n*ode labels and *a*rc labels and we use $N(n|q)$ for the probability of node label n at tree location q according to DP N (similarly for A). We further overload our notation to use $n(node)$ to indicate the node label for a given node,

1 Alternatively, one could define a separate process for selecting state labels for frontier nodes, but we leave this to future work.

$a(node)$ to indicate the outward-going arc labels from *node*, and $q(e)$ or $q(node)$ to indicate the location of a given subtree or node within the tree as an (n, a) pair. root(e) is a function selecting the root node of an elementary tree e and children$(node)$ indicates the child subtrees of a given node.

The prior over elementary trees $P(e|q)$ supposes that each node's label is influenced by the label of its parent as well as the label of the arc from its parent to itself. This makes sense because, for example, the set of entities which can appear as the subject of a sentence depends in part on the verb which forms the root of that sentence. Roughly speaking, we might expect different kinds of Arg0s for the verb HAVE than the verb BE.

For the arc labels, however, we condition only on the label of their source node, because the trees are built top down and we similarly expect different words to have different kinds of children. For example, the broad coverage grammar of OpenCCG (cf. Section 6.2.1) uses the arc labels FIRST and NEXT only for conjunctions like 'and', 'but', and ',' (i.e. a comma).

The distributions over node labels given tree locations $N|q$ and arc labels given source node labels $A|n$ are DPs over simple uniform priors:

$$N|q \sim \text{DP}(1.0, \text{Uniform}(\{n \in corpus\})) \tag{41}$$

$$A|n \sim \text{DP}(1.0, \text{Uniform}(\{a \in corpus\})) \tag{42}$$

$$\tag{43}$$

These simple priors ensure that there is some probability of seeing any node or arc label at any given position in the tree but does not make any further assumptions about where they might occur. This simplifies our model, by limiting the number of conditional distributions we need to fit, and we rely on the Dirichlet Processes to model the true distribution.

In the remainder of this thesis I use subscripts TP and LF on these distributions to indicate whether they are associated with the grammars for the text plans or the logical forms, respectively. Accordingly the corpus in Equations 41 and 42 consists only of TP or LF trees, depending on the grammar.

9.2 HIERARCHICAL CRP FOR STSG DERIVATIONS

Our Synchronous Tree Substitution Grammar (sTSG) model has two additional distributions: (1) a distribution over pairs of TP and LF elementary trees; and (2) a distribution over pairs of tree locations representing the probability of those locations being aligned to each other.

Similarly to the generative story for a single TSG, we begin by sampling a pair of TP & LF elementary trees, a *TreePair*, for the root of

Algorithm 1 Simple generative story for strictly synchronous TSGs

alignments ← (*None*, *None*), (*None*, *None*)
while *alignments* **do**
  alignment ← *pop*(*alignments*)
  Sample a new TP-LF etree pair *TreePair* for the given alignment.
  Sample a new set of alignments $alignments_i$ for the frontier nodes (i.e. substitution sites) of *TreePair*.
  for *alignment* in $alignments_i$ **do** *alignments*.append(*alignment*)
  end for
end while

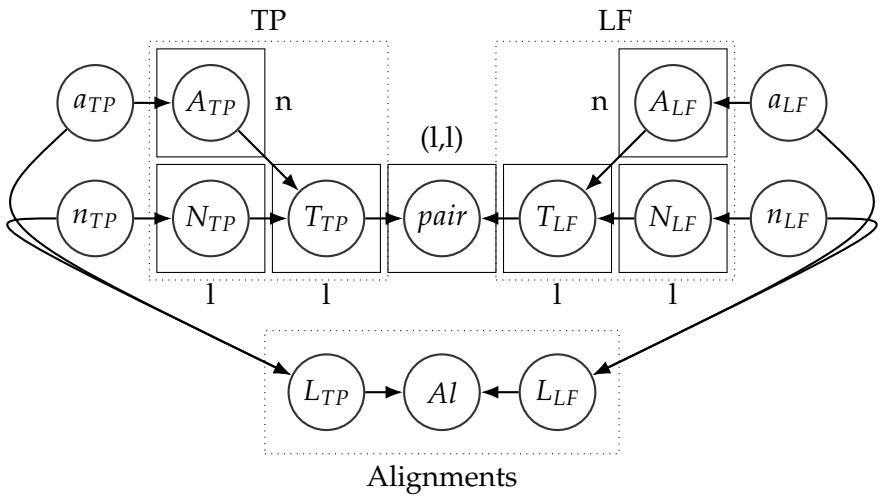


Figure 29: Dependencies in our statistical model. Nodes labeled with a or n represent parameters, all other nodes represent Dirichlet processes over base distributions with $\alpha = 1$. Here n indexes node labels for TPs or LFs as appropriate, while l similarly represents tree locations.

the derivation (cf. Algorithm 1). We then sample alignments for the frontier nodes of the TP to the frontier nodes of the LF. For each of these alignments, we then sample the next TreePair in the derivation and repeat this sampling procedure until no unfilled frontier nodes remain.

The distribution over TreePairs for a given pair of tree locations is given by a Dirichlet process with a simple prior which multiplies the probability of a given TP elementary tree by the probability of a given LF elementary tree (where these probabilities are as defined in Equation 39):

$$pair|l_{TP}, l_{LF} \sim \mathrm{DP}(1.0, P(e_{TP}, e_{LF}|l_{TP}, l_{LF})) \tag{44}$$

$$P(e_{TP}, e_{LF}|l_{TP}, l_{LF}) = T_{TP}(e_{TP}|l_{TP})T_{LF}(e_{LF}|l_{LF}) \tag{45}$$

In this very simple model, text plans and logical forms are treated as independent in the base distribution and we rely on their joint

observation to reveal the relationship between our semantic and morphosyntactic representations. A natural extension would be to replace this prior with one which already conditions the LF on the TP elementary trees, however the implementation of such models is more complicated, so we leave this improvement to future work.

The distribution over possible alignments is given by a hierarchical Dirichlet process, which uses the product of the probability of the TP substitution site with the probability of the LF substitution site as a prior. Each of these probabilities is given by a Dirichlet process over a uniform prior for possible tree locations in the respective grammar:

$$Al \sim \mathrm{DP}(1.0, P(q_{TP}, q_{LF})) \tag{46}$$

$$P(q_{TP}, q_{LF}) = P(q_{TP})P(q_{LF}) \tag{47}$$

$$P(q_{TP}) \sim \mathrm{DP}(1.0, \mathrm{Uniform}(\{q_{TP}\})) \tag{48}$$

$$P(q_{LF}) \sim \mathrm{DP}(1.0, \mathrm{Uniform}(\{q_{LF}\}))$$

9.3 GIBBS OPERATORS FOR SAMPLING STSG DERIVATIONS

Our Gibbs sampler adapts a blocked sampling approach from TSG induction for simple Phrase Structure Grammar (PSG)-based trees (cf. Section 5.1) to synchronous TSGs over the labeled dependency graphs (cf. Section 3.2). This updated sampling regime uses two Gibbs operators throughout: `split-and-align` and `sliding-target` alignment. A third operator (`consider-roots`) is added in some models.

9.3.1 *Split-and-align*

For each position in the TP of a given TreePair, we resample the *derivation type* of that node based on the probability of the resulting TreePairs. For example, if a node is currently an 'interior' node, not aligned to any specific node in the LF, then the probability of it remaining an interior node is proportional to the joint probability of its dominating substitution site and the LF etree to which it is aligned. The probability of this node becoming a substitution site, then, depends on which nodes in the LF it might be aligned to. For each node in the LF to which the current TP node could align while remaining consistent with a TSG, we calculate the probability of the TreePairs that would result from this split and alignment. We then sample the split-merge decision for this TP node jointly with its alignment, by normalizing over the possible options.

The `split-and-align` operator allows us to explore new elementary trees farther away from our initial alignments.

9.3.2 *Sliding alignments*

Under the `sliding-target` operator, we consider each alignment in a TreePair and consider the possibility that there is a small error in the alignment, asking whether we should choose the LF node above or below the current LF node associated with a given TP node. For this operation, we assume that the current split of the TP tree is correct, so we sample the decision to move the alignment according to the probability of the resulting TreePairs and alignments.

9.3.3 *Adding root alignments*

CONSIDERROOTS & WITHROOTS

Because large TreePairs have a low probability in the base distribution, instances of large trees in the initial alignments are difficult to unlearn. Since our initial alignments are heuristic, they have a tendency to provide most alignments near the leaves of the trees. This means that large elementary TreePairs are especially common at the root of a TreePair.

Therefore we explored a setting in which the roots of the TP and the LF trees are initially unaligned and the model only considers creating an elementary TreePair for the root of a TreePair if it would be small enough (CONSIDERROOTS). In practice, we set the initial criteria to only add elementary TreePairs at the root if the TP etree contained only two nodes. For each 100 iterations, we increased the allowed size of the tree by one node, so that if a large root elementary TreePair is necessary it can eventually be added to the model, after more evidence for other rules has been accumulated without early interference from over-large elementary TreePairs.[2]

Naturally, we also explored what would happen when the roots are left in throughout the entire sampling procedure (WITHROOTS).

9.4 TRAINING THE MODEL

We first conducted experiments using the SPaRKy Restaurant Corpus (SRC) and then using the Extended SPaRKy Restaurant Corpus (ESRC) (see Chapter 7 for descriptions of these corpora). These experiments varied along several dimensions, exploring different ways of generating initial alignments between TPs and LFs, different ways of initializing the model from these alignments, different treatment of the root nodes, and different values of the parameter α.

2 Note that we use `consider-roots` to refer to the Gibbs operator and CONSIDERROOTS to refer to models which include this operator during training.

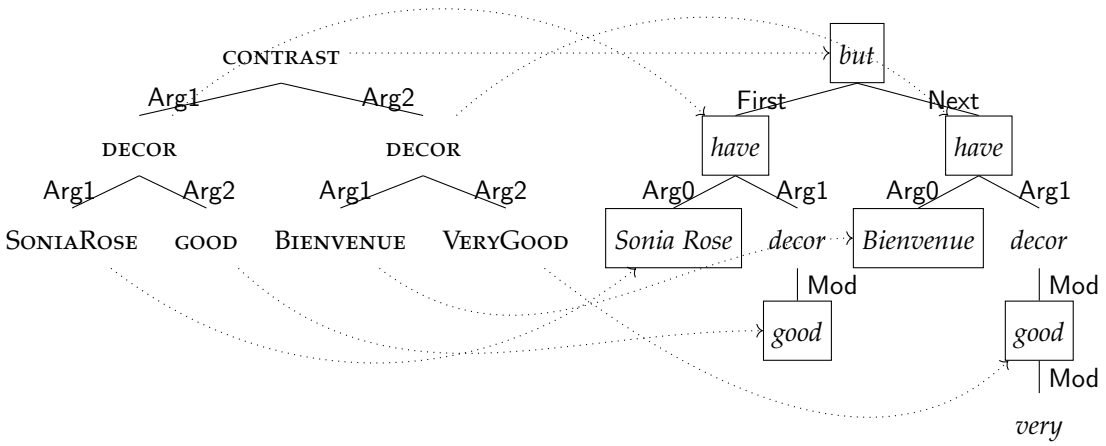


Figure 30: Example TreePair showing a text plan (left) and a 'logical form' (right). Arrows run from TP substitution sites to LF substitution sites, which are boxed. Some morphosyntactic details suppressed for readability.

9.4.1 *Initializing alignments*

Rather than randomly splitting the TreePairs in the training corpus into sTSG rules, we initialized our model using heuristic alignments developed in the course of a Bachelor's thesis (Pietsch, 2017).

We applied the heuristics in two settings. In the first, we used the text plans and logical forms as they are, allowing the pre-terminal nodes in the text plans to be aligned to nodes in the logical forms based on an 'exact string match' criterion. The pre-terminal TP nodes represent predicates, but are nonetheless aligned to nouns in the LF which do not dominate both arguments of the TP predicate.For this reason we also generated alignments after renaming these pre-terminal nodes in the TPs to include a standard prefix ASSERT_ to guarantee that they would not be aligned to these nodes on the basis of this criterion. For example, this would correspond to the TP DECOR nodes in Figure 30 becoming ASSERT_DECOR so that they are not inadvertently aligned to the syntactic nodes labelled *decor*.

We explored two different approaches to post-processing the heuristic alignments. The first was designed to minimally ensure that the alignments were bijective, pruning cases of where a single node in one tree was aligned to more than one node in the other. Because this metric mostly preserves the matches and least common ancestor (LCA) identifications from Pietsch (2017), we call this the MATCH+LCA model. For the second we created manual node-level alignments for a subset of the dev set and then created more careful post-processing rules to improve F-score; we name this the TARGETED model.

Several specific weaknesses motivated the creation of the TARGETED model. Firstly, the reliance on exact string matches led the first phase

Heuristic	P	R	F
MATCH+LCA	77	49	66
TARGETED	86	52	72

Table 13: Accuracy of the two heuristics for post-processing initial alignments with respect to gold-standard sTSG alignments.

of alignment to often produce disconnected subtrees that could not be joined by the LCA alignment.

For example, consider the nodes labelled *decor* in Figure 30. If we take the children of a DECOR node in the TP, we see that their LCA (i.e. DECOR) must be aligned to one of the *have* nodes in the LF; however, a string matching alignment would always align to the Arg1 of one of the *have* nodes, preventing a better initial alignment.

Secondly, the implementation included aggressive pruning for individual nodes when their sibling from the TP could not also be aligned via exact string matching. For example, VERYGOOD is split across two nodes in the LF in Figure 30. In this case, the heuristic aligner drops the alignment for *Bienvenue*, despite the exact string match.

We updated the heuristic to address these issues and evaluated the ability of the two heuristics to identify optimal alignments for an sTSG. To do this we manually annotated 10 TreePairs from the development section of our dataset with an optimal segmentation for sTSG rules.This resulted in 93 alignments between nodes in the TPs and their corresponding LFs.[3]

Table 13 shows the precision, recall, and F1-score for the MATCH+LCA and TARGETED models. The result was a clear improvement in the degree to which the initial alignment corresponded to our notion of optimal sTSG rules.[4]

9.4.2 *Initializing the model*

When initializing a statistical model of this kind, the standard practice is to first read the random initialization of the corpus into the model. That is, we use the initial alignments proposed by our heuristics and pre-processing as paired substitution sites in a set of sTSG derivations for the corpus. Each TreePair in this DEFAULT INITIALIZATION is 'observed' by the model, adding to the counts in our hierarchical Chinese Restaurant Processes.

The initial state of our corpus, though based on heuristics, can include a wide variety of TreePairs which are suboptimal, including overly specific and overly large trees as well as (potential) misalign-

3 These TPs had a total of 129 nodes; the LFs had 244.

4 To verify that it was worth the effort spent on the heuristic, we also calculated the F1-score for a model which only aligns the roots and any unique string matches at the leaves: P = 100, R = 39, and F = 56.

ments. Because we want our model to 'forget' any detrimental elements present in this initialization, we also explored a SAMPLING INITIALIZATION. In this mode the initial alignments are treated as the corpus state from the 'previous iteration' of Gibbs sampling and we begin immediately with Gibbs sampling without any special initialization.

9.4.3 *Training settings*

We trained our primary models for evaluation for 10k iterations, each iteration including both the `split-and-align` and `sliding-target` operations. In the CONSIDERROOTS condition each iteration also included an extra Gibbs operation to check whether the root TreePair should be added to the grammar or not.

In order to evaluate other aspects of the model, such as convergence-like behavior after n iterations and variation due to random seeds, we also trained a larger set of models for 2k iterations.

9.5 AUTOMATED METRICS ON THE ORIGINAL SRC

We conducted several experiments to understand the properties of our model. First we looked at how much the choice of random seed affected the output of our model. We then looked at the impact of 4 different initializations across 3 different alpha values with both modes of model initialization and both treatments of roots, resulting in a comparison of 48 model variants.

In these experiments we sought to understand how the model varies with respect to generalization based on coverage of the dev set and how these variations affect the resulting texts.

9.5.1 *Model variation by random seed*

While it is becoming increasingly common to report system performance based on averages across five runs in the neural network literature, this is not necessarily enough measurements to be representative of the mean system performance.[5]Therefore we conducted an experiment across 30 different random seeds, training 4 instantiations of the model for 2k iterations with each of the 30 seeds. To understand the impact of the random seeds, we look at the differences in dev set coverage (how many TPs can we successfully generate an LF for?) and the resulting texts.

We found that the MATCH+LCA initialization post-processing *without* ASSERT_ insertions performed best, although we also found that

5 It has even been remarked that some published results seem to be impossible to replicate without knowing the exact random seed that the original authors used.

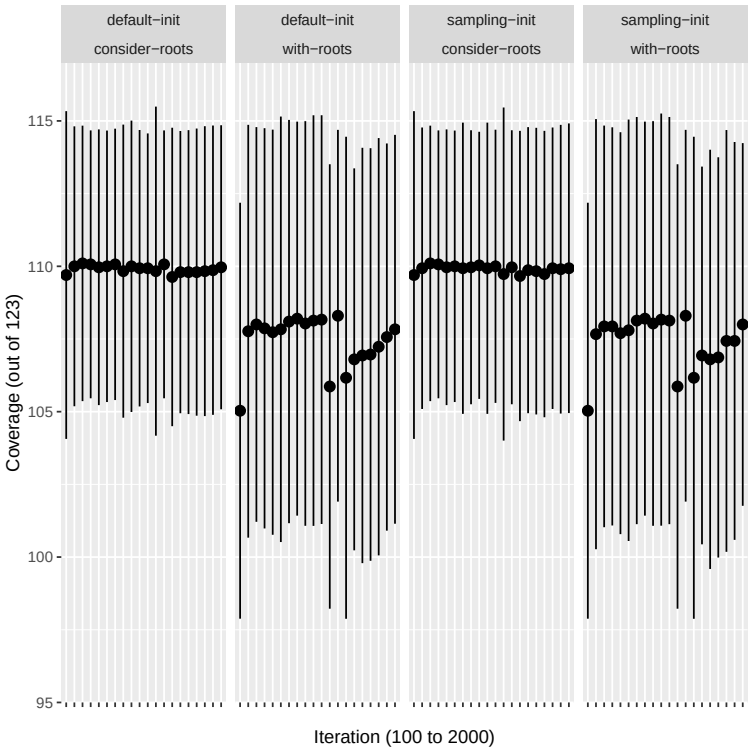


Figure 31: Plot of dev set coverage for four different `bn4nlg` models (30 random seeds each). Note that the difference in initialization (default v. sampling) is almost non-existent, while there is substantial difference between the WITHROOTS and CONSIDERROOTS models.

the choice of random seed could have a larger impact on model performance than the details of the model chosen.

9.5.1.1 *Dev set coverage*

In Figure 31 we plot the mean coverage (with 95% confidence intervals) on the development set for these four models.. These results show that the two modes of initialization (sampling vs. simply reading off the initial alignments) produces no difference in the model performance on this metric. We can also see that the models WITHROOTS begin with lower coverage than the CONSIDERROOTS models, although these differences diminish over the course of the 2k iterations used in this evaluation.

These confidence intervals represent estimates of the mean performance of these models. This is useful for assessing how well one model performs compared to another, but also allows us to gauge the impact of the random seed on model performance. Based on these figures we expect the coverage of the CONSIDERROOTS models to be higher on average than the coverage for the WITHROOTS models. The range of values is approximately the mean plus-or-minus 5 text plans, meaning that the best performing random seed for one model versus

the worst performing random seed can result in a swing of 8% ($= \frac{10}{123}$) in coverage.

Given that the difference in the means is only about 5, this suggests that the best performing instances of the weaker models can outperform many instances of the stronger models, depending on the choice of random seed. Ideally we would like our systems to be robust to variation in the random seed, rather than having to run a model many times to choose the 'best' random seed.

Most importantly, this analysis suggests that we need to do more work in general to document the sensitivity of our models to initial conditions than is typically done in the NLP community, at least if we are interested in general recommendations and not only best-case performance.

9.5.1.2 *Text differences*

Beyond differences in coverage, it is important to understand how much the random seed influences the actual texts produced by the model.

Figure 32 shows a 'confusion matrix' of BLEU scores comparing the output of one model with the output of another. Note that changing the random seed dramatically changes the text produced: almost every BLEU score is between 50 and 62, except for those along the diagonal (comparing a text to itself). Table 14 shows two texts from a single model across 10 random seeds for a qualitative impression of what these differences mean.

9.5.2 *Model variation by parameter settings*

In these explorations we hold the random seed constant and look at the effect of different parameters on the model's coverage. In particular, we systematically vary the preprocessing (MATCH+LCA vs. TARGETED), whether or not the ASSERT_ rewriting was performed before initializing the alignment, the method of model initialization (DEFAULT or SAMPLING), at what point root nodes where added to the model, and different values of α for the CRPs.

9.5.2.1 *Dev set coverage*

As the graphs in Figure 33 indicate, the choice of α and the choice between how to initialize the model does not have a large impact on dev coverage. However, we find that the MATCH+LCA heuristic initialization *without* ASSERT_ rewrites performs substantially better than the TARGETED initialization (with or without ASSERT_ rewrites) and than the MATCH+LCA initialization with ASSERT_ rewrites.

Moreover, we observe that including the roots from model initialization onward (WITHROOTS) results in lower coverage than consider-

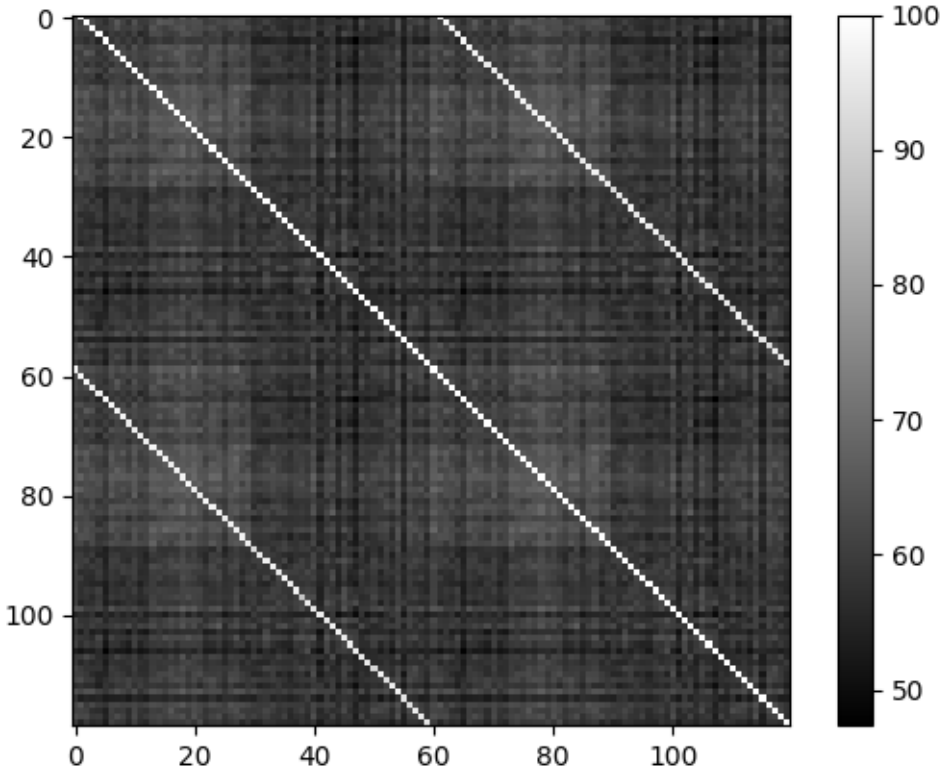
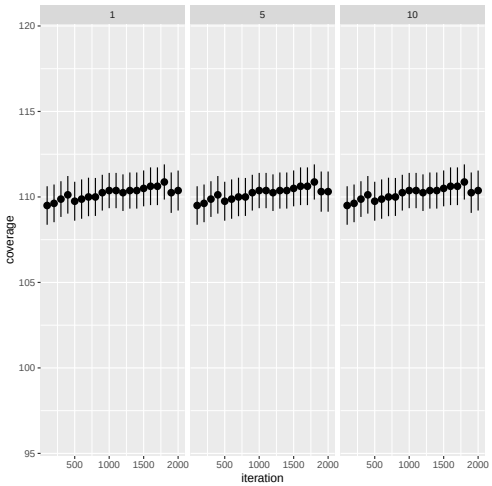


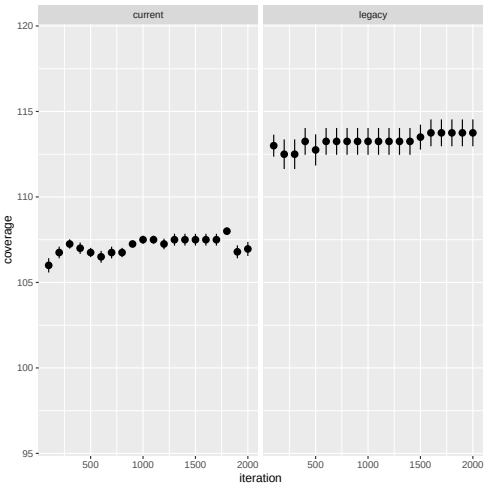
Figure 32: Heatmap of BLEU scores for different random seeds on the SRC dev set. The bright diagonal lines correspond to (1) perfect self-similarity and (2) high similarity between the DEFAULT initialization and SAMPLING initialization for the same random seed. Otherwise, texts from different random seeds were very different from each other. The first 60 rows (columns) of the figure correspond to the DEFAULT initialization and the next 60 to the SAMPLING initialization. Within each quadrant, the first 30 rows (columns) correspond to CONSIDERROOTS and the next 30 to WITHROOTS. The 30 points within each of these regions each correspond to a different random seed.

ChezJosephine has the best overall quality among the selected restaurants . it has very good service and it has food food quality . it has very good decor . (x2)
ChezJosephine has the best overall quality among the selected restaurants . it has good decor and it has food food quality . it has very good decor .
ChezJosephine has the best overall quality among the selected restaurants . it has very good service . it has very good decor .
ChezJosephine has the best overall quality among the selected restaurants . and food food quality it has good food quality . it has very good decor .
ChezJosephine has the best overall quality among the selected restaurants . it has food food quality and it has food food quality . it has very good decor . (x2)
ChezJosephine has the best overall quality among the selected restaurants . very good service , and it has food food quality . it has very good decor .
ChezJosephine has the best overall quality among the selected restaurants . it is a Japanese , Latin American restaurant , with food food quality . it has very good decor .
ChezJosephine has the best overall quality among the selected restaurants . it has very good decor and it has very good food quality . it has very good decor .
Monsoon 's price is 26 dollars . however , LemongrassGrill 's price is 22 dollars . Monsoon is a Vietnamese restaurant but LemongrassGrill is a Thai restaurant . (x6)
Monsoon 's price is 26 dollars . however , LemongrassGrill 's price is 22 dollars . Monsoon 's an Vietnamese restaurant . LemongrassGrill , however , is a Thai restaurant .
Monsoon 's price is 26 dollars . however , LemongrassGrill 's price is 22 dollars . Monsoon 's an Vietnamese restaurant . on the other hand , LemongrassGrill is a Thai restaurant . (x2)
Monsoon 's price is 26 dollars . however , LemongrassGrill 's price is 22 dollars . Monsoon 's an Vietnamese restaurant . on the other hand , LemongrassGrill is a Thai restaurant . Monsoon is a Japanese , Sushi restaurant while LemongrassGrill is a Japanese , Vegetarian restaurant .

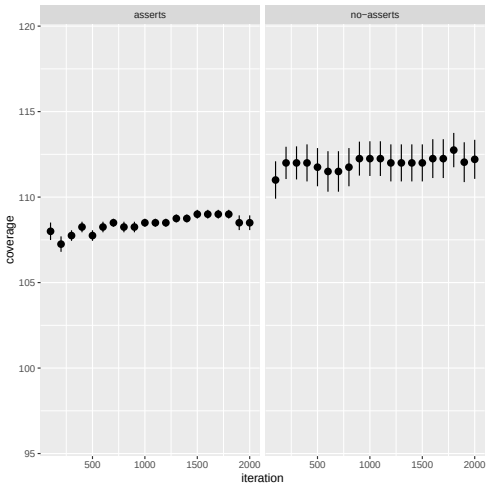
Table 14: Texts produced by `bn4nlg` using the MATCH+LCA initialisation without ASSERT_ insertions, using the DEFAULT initialization and using the `consider-roots` Gibbs operator. We sampled 10 texts each for two textplans; repeated texts indicated with (xN) at the end of the text, indicating how many times that text appeared in the sample. Sentence Bilingual Evaluation Understudy (BLEU) range from 65 to 100 comparing each sentence in the two sets to all the other texts in that set.



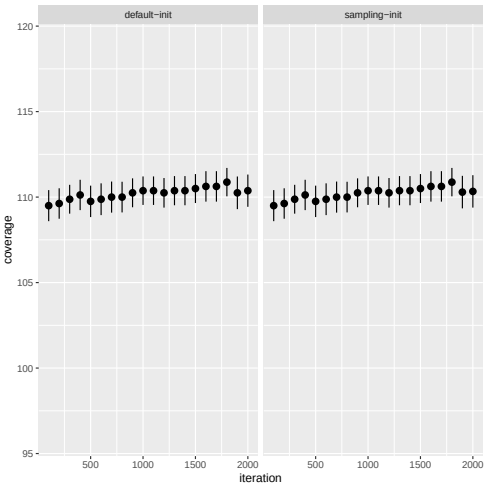
(a) Dev coverage for $\alpha = 1, 5, 10$, averaged across all other parameter values.



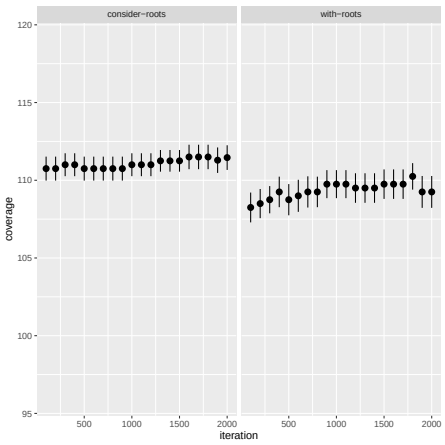
(b) Dev coverage for MATCH+LCA and TARGETED preprocessing, averaged across all other parameter values.



(c) Dev coverage for using assert rewrites or not, averaged across all other parameter values.



(d) Dev coverage for default vs. sampling model initialization, averaged across all other parameter values.



(e) CONSIDERROOTS vs. WITHROOTS, averaged across all other parameter values.

Figure 33: Dev set coverage on the SRC for different parameter settings.

ing them later (considerRoots), consistent with our findings in the previous subsection (Sec. 9.5.1.1).

9.5.2.2 *Text differences*

Figure 34 shows a 'confusion matrix' where BLEU scores are represented by brightness, as in Sec. 9.5.1.2. The two large blocks represent the targeted and match+LCA initializations, so we can see that they result in substantially different texts. Within each of these blocks, the four-by-four grid of blocks is alternating between the considerRoots and withRoots conditions, revealing large differences in the texts resulting in these two conditions. However, the checkerboard pattern confirms what we observed in the preceding section: the differences between the sampling and default initializations is minimal. Finally, the groups of 3×3 within this checkerboard correspond to different α values, showing that different choices of α does not result in substantially different texts.

9.5.3 *Discussion*

Our automated evaluations provide some initial insight into how the system performs. Some model choices have a large influence on the text quality (namely, choices relating to pre-processing initial alignments and the handling of root nodes) while others do not substantially impact the resulting texts (namely, the choice of α value or the way in which the model was initialized). In some cases the choice of random seed has a larger impact on generalizability than the choice of model, and a different random seed always results in substantially different texts.

9.6 HUMAN EVALUATION ON THE SRC

For the human evaluation we wanted to evaluate how well our texts achieve their intended goals of expressing semantic structure and discourse cues in a natural way while also comparing to another NLG system and the original corpus texts. For the alternative system, we use a neural NLG system (TGen), which we compare to two instantiations of our model (described in Sec. 9.6.2).

Subjects provided feedback about text quality using a new interface for NLG evaluation (described in Sec. 8.4.2.2). Overall we found that our model can achieve better semantic control with a similar degree of fluency compared to the baseline.

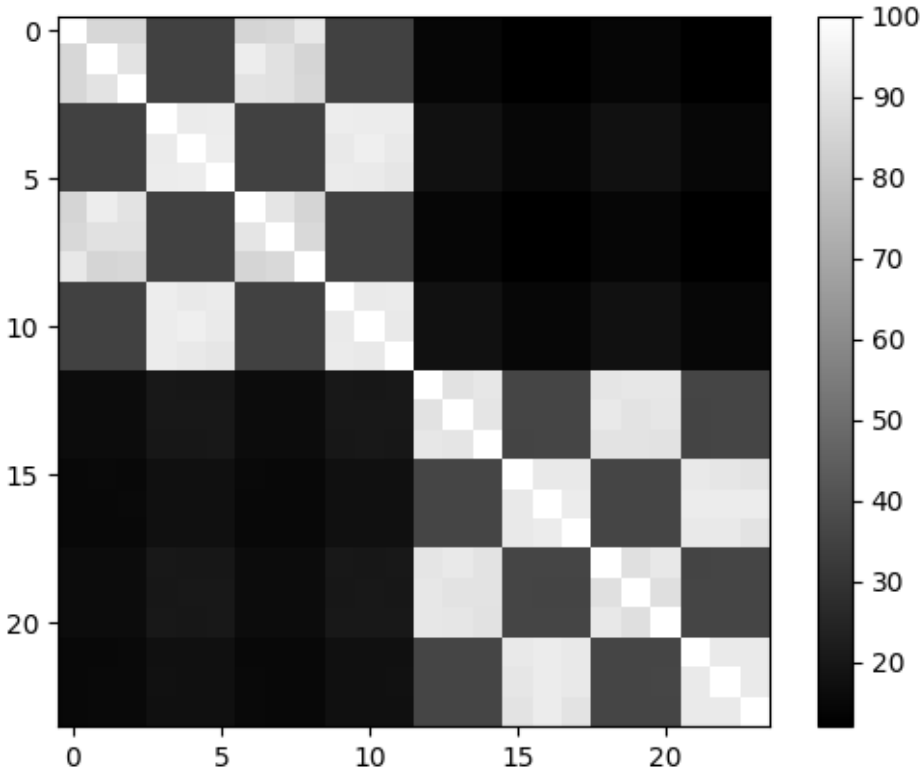


Figure 34: Heatmap of BLEU scores for different parameter settings. The first 12 blocks correspond to the TARGETED initialization and second 12 to the MATCH+LCA initializations. The 4×4 grids of blocks alternate between the CONSIDERROOTS and WITHROOTS conditions in blocks of 3. Each of the blocks of 3 correspond to the 3 α values we tested. Lighter colors indicate higher similarity.

9.6.1 *Baseline model*

Dušek & Jurčíček's (2016) TGen system provides the baseline for our evaluation. This neural NLG system uses a flat meaning representation (cf. Section 7.2) consisting of a sequence of slot-value pairs coupled with corpus texts to train a sequence-to-sequence (seq2seq) model with attention.Despite the apparent simplicity of this model's architecture, it served as an extremely competitive baseline in the End-to-End Generation Challenge, with only one system surpassing it in each of the two human evaluations (overall text quality & naturalness) (Dušek, Novikova & Rieser, 2020). TGen is readily available and easily adapted to new datasets, making it an ideal neural baseline for comparison.

Because TGen by default reorders the input meaning representation into a standard order, we had to disable this setting to make TGen more competitive with our model when it comes to presenting facts in the intended order.

TGen also expects each text to describe only a single restaurant (or other entity) and is not equipped to make use of reference annotations of the sort we introduced in our annotation scheme (i.e. RefDAs). Half of our texts recommend a single restaurant, allowing us to use TGen as intended without any additional preprocessing. For those texts which compare multiple restaurants, we had to split up the texts and meaning representations by restaurant. We did this using a heuristic which splits the text each time a different restaurant is mentioned by name and grouping the slot-value pairs with each accordingly. At training time, each MR-text pair was given to TGen as a separate training instance, and at testing time we used similarly chunked MR inputs and then concatenated the resulting texts.

TGen uses a beam during generation with a semantic completeness scorer trained simultaneously with the generation system to rerank possible outputs. This scorer is designed to ensure that the highest ranked text for each input also includes all and only the facts given in that input. In order to perform a fair comparison between our system and TGen, we use the same semantic completeness scorer on the 100-best[6] outputs according to the rules learned by our system.

9.6.2 *Choosing instantiations of `bn4nlg` to compare*

Based on the findings of the previous section, we chose to compare two instantiations of `bn4nlg` using the 'best' settings available but differing with respect to heuristic initialization (i.e. MATCH+LCA vs. TARGETED). We set $\alpha = 1$ throughout, did not use `assert` rewrites,

6 We chose to do reranking over the 100-best list based on TGen's choice of beam width = 100.

stuck to the default initialization, and only considered roots later in the sampling process.

9.6.3 *Evaluation methods*

We recruited subjects using Prolific Academic[7] to evaluate the fluency and adequacy of our texts. Each subject took a mean of 36:08 minutes (stddev 7:18) to evaluate 20 texts drawn from a set of control texts, the baseline system, or our system. Subjects ranged in age from 19 to 55 (median 25, mean 28, stddev 9.0) from a variety of socioeconomic backgrounds and English speaking countries and were paid 3 GBP for their participation.

9.6.4 *Results*

9.6.5 *Semantic Fidelity*

Table 15 shows the results of our investigation into semantic adequacy. As expected, the original corpus does not include any omissions, although we did receive 8 notes that some meaning was added. Examining these cases, we found that these were errors on the part of the subjects peforming the evaluation: they appear to have interpreted the prompt ('Does the text include any extra details?') as asking whether there was any information which they would consider superfluous rather than whether there was any information *not present in the meaning representation* which was added to the text.[8]

We see that the TARGETED `bn4nlg` system performs the worst on semantic fidelity, while the MATCH+LCA model performs substantially better than TGEN, reducing the overall number of texts with missing information and having about two-thirds as many facts omitted overall.

Figure 35 shows the permutation distance[9] between the order of the facts given by the source TP and the order of the facts in the resulting text as indicated by our participants. Subjects mostly agree that the corpus is in the correct order, which we know to be objectively true. Both `bn4nlg` models appear to do a much better job of preserving the intended order of mention for facts than TGEN.

7 https://www.prolific.ac

8 This error highlights the importance of repeating *complete* questions in the survey proper, and not relying on subjects to remember the detailed instructions given during training. Of the 25 subjects recruited for this task, one systematically interpreted this question this way while one other appears to have used this interpretation at least 3 times.

9 We use Kendall's tau as implemented by Irurozki, Calvo & Lozano, 2016 for all permutation distance measures reported in this thesis.

System	Miss.	1	2	3	Added
Corpus	0	0	0	0	8
TGen	66	28	16	2	17
match+LCA	43	43	0	0	18
targeted	71	45	12	2	37

Table 15: Semantic fidelity in the first experiment. # of facts dropped by each system (out of 630), and # of instances (out of 125 possible) where a text was missing 1, 2, or 3 facts. No texts dropped more than 3 facts. The last column is the number of instances where a text was marked as having ≥ 1 inserted facts.

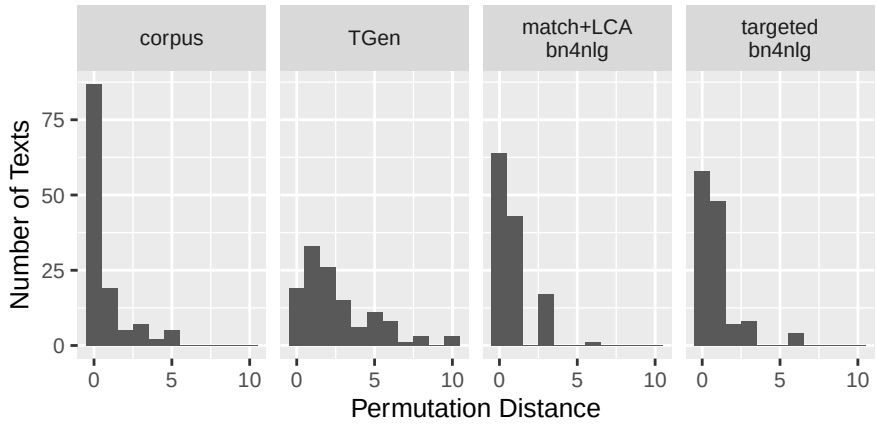


Figure 35: Frequency of different permutation distances for each system in our first experiment.

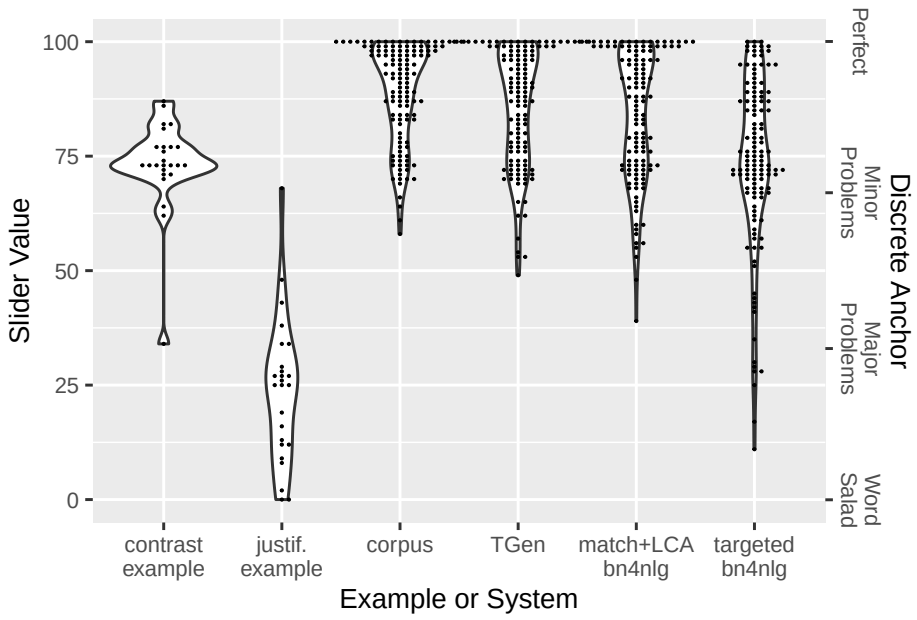


Figure 36: Fluency ratings for the instructional texts and each of the systems evaluated in our first experiment.

9.6.6 *Fluency*

The first two columns of Figure 36 show how subjects used the scale for the instructional example texts (one expressing contrast and one expressing justification). From the final four columns, we see that the model trained using the TARGETED initialization has the only obviously different distribution of fluency ratings.

Figure 37 shows the scores for each system, broken out by text.We see that the corpus texts, MATCH+LCA `bn4nlg` texts, and TGEN texts are all rated similarly for fluency, while the TARGETED `bn4nlg` system performs worse.

A post-hoc Wilcoxon signed-rank test with Bonferroni correction supports this finding: the mean rating for the TARGETED initialization (73.7) is significantly different from the corpus texts (89.2; $p < 10^{-12}$), the MATCH+LCA initialization (84.0; $p < 10^{-5}$), and TGEN (85.9; $p < 10^{-7}$).[10]

We also found that subjects varied both in terms of the range of values used on the fluency scale and in terms of how continuously or discretely they used the rating scale, as shown in Figure 38. While most participants clearly used the scale in a continuous fashion, several participants (e. g. 4, 5, 22) almost always chose points very close to the text based anchors we provided (see right-hand labels for the scores).

10 No other significant differences at corrected $\alpha = 0.0083$

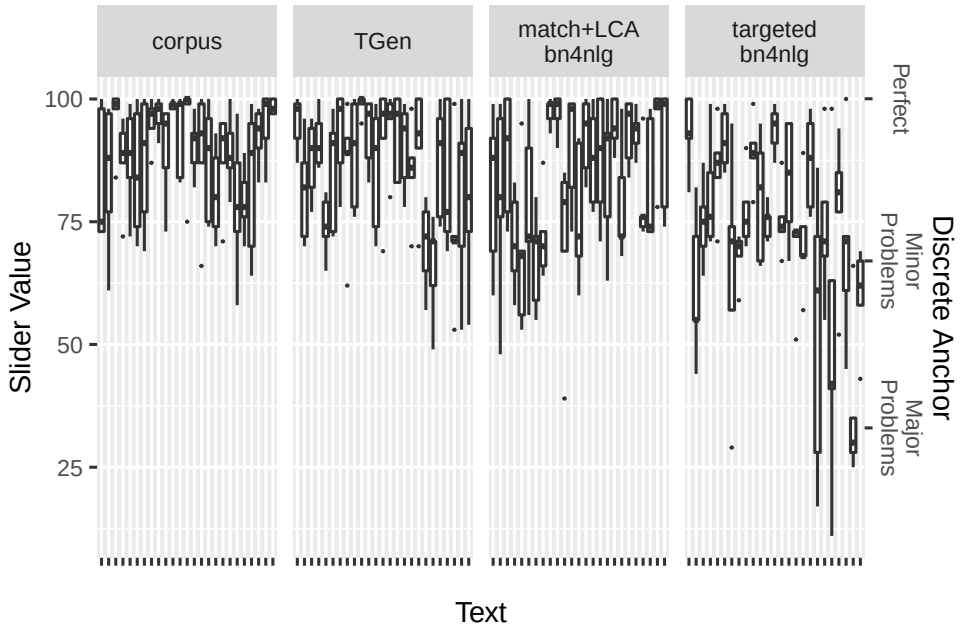


Figure 37: Boxplot of ratings for each item rated in the first experiment, split out by system. Ratings for the corpus texts appear to be the most consistent, with large disagreements about the quality of texts produced by TARGETED `bn4nlg`.

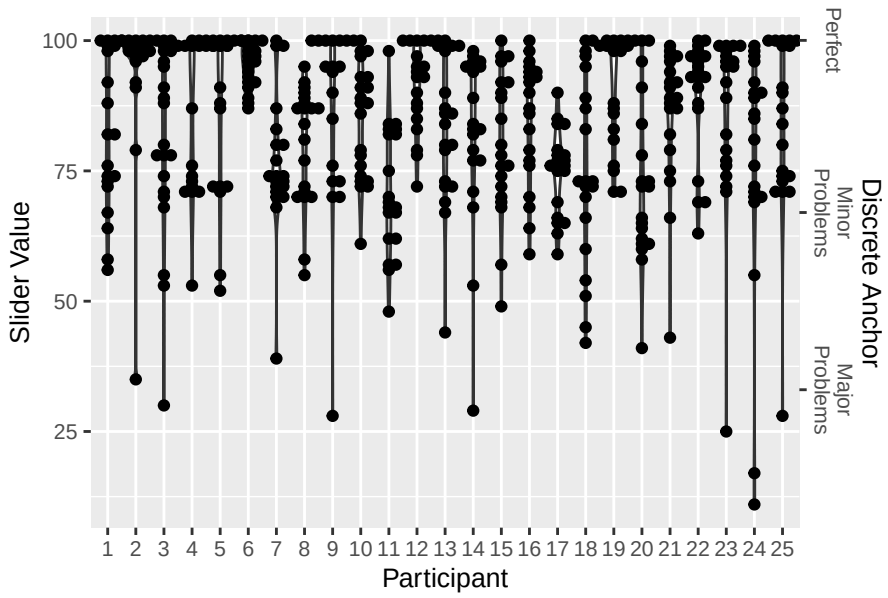


Figure 38: Fluency results for the first experiment, broken out by individual participant, showing different patterns of usage by individuals.

9.6.6.1 *Discourse*

We discovered during evaluation that the test set contains mostly JUSTIFICATION relations, so we focus only on this analysis here. Out of 120 possible instances, subjects identified JUSTIFICATION 74 times in the corpus texts, compared to only 67 instances for both TGEN and the MATCH+LCA `bn4nlg` model. Again, the TARGETED model performed substantially worse, being perceived as expressing justification only 57 times.

9.6.7 *Discussion*

Our human evaluation shows that the model using the MATCH+LCA heuristic alignment postprocessing outperforms a strong neural baseline with respect to semantic fidelity while preserving perceived fluency.

Though we designed the TARGETED post-processing to provide initial alignments that match those we would expect to find in a handwritten sTSG for this dataset, the model with TARGETED post-processing results in worse performance for all of our human evaluations and lower coverage on the development set (cf. Figure 33). The original improvements in our TARGETED post-processing over MATCH+LCA were mostly due to improvements in precision rather than recall (cf. Table 13). Taken together this suggests that the lower precision of the MATCH+LCA postprocessing actually resulted in useful noise by creating smaller trees, which would increase the dev set coverage, allowing the models trained from those initial alignments to discover more reusable sTSG rules. This would also introduce a variety of incorrect rules, but these must have been lower in frequency relative to the rules that ended up being used to generate texts for our human evaluations. Therefore it may be better in general to more strongly bias the model toward creating small rules rather than small derivation sequences (cf. discussion in Section 4.6).

Unfortunately, the data split used for these experiments did not include enough instances of the CONTRAST relation to draw strong conclusions about the performance of our model with respect to the expression of discourse relations. For this reason, we propose a new test set in Section 9.7.

9.7 AN IMPROVED TEST SET FOR CONTRAST

Our approach to generation is intended to provide better discourse-level control, so it is important for us to evaluate on a test set which contains both CONTRAST and JUSTIFICATION relations in sufficient numbers. For this evaluation we generate a new set of text plans, which we call the NOVELCONTRAST datset. Since these text plans were not

originally present in the SRC, we can no longer compare directly to that corpus but instead can only compare among different instances of our model and our baseline.

To generate new text plans, we used the same groupings ('tasks') of restaurants used in the SRC.[11] First the script samples a 'task', then a restaurant from the task, and then a second restaurant from that task which has some of the same properties as the first restaurant.[12] Then it samples a subset of the shared properties of the two restaurants to be contrasted. The last stage of selection is to determine which properties have the same values (e.g. both restaurants have 'good' decor) and which properties have different values.

With the restaurants chosen, properties-to-mention chosen, and their similarities & differences identified, the script then samples the choice of TP-style: serial or back-and-forth (cf. Sections 2.1.2 and 7.3.2.1). Serial text plans have a root CONTRAST node with INFER children, with each of these INFER nodes dominating the properties of a given restaurant. Back-and-forth text plans have an INFER node at their root, with INFER and CONTRAST children. For each property-to-mention, the node dominating the individual property assertions is an INFER node if the property has the same value and a CONTRAST node if the values differ.

This represents a very simple text planning algorithm; in an actual pipeline, we would expect to encounter more varied textplans containing the CONTRAST relation. Indeed, this algorithm does not try to add additional structure to group sibling INFER nodes into subtrees, which would result in more binary and ternary branching and fewer nodes with more than ternary branching.

We see the impact of this in processing the text plans we generate. We generated 50 new datasets consisting of 100 TPs, with each dataset using a different random seed. Applying the rules used in our previous evaluation of the MATCH+LCA and TARGETED models, we only generated LFs for 30.5% of each dataset on average. Figure 39 shows a histogram of these coverage scores.

This highlights a difficulty in our system's ability to generalize to new input text plans, likely due to rules learnt by `bn4nlg` containing more lexicalization than necessary and therefore being overly specific to particular combinations of entities and values observed in the training data.

9.7.1 *Human evaluation for* CONTRAST

Despite low levels of coverage, we can still assess the quality of the texts that we are able to generate. To ensure a large and varied enough

11 Participants in Walker et al. (2007)'s were tasked with finding restaurants based on their value, cuisine type, and/or location.

12 All sampling mentioned in this section is from a uniform distribution.

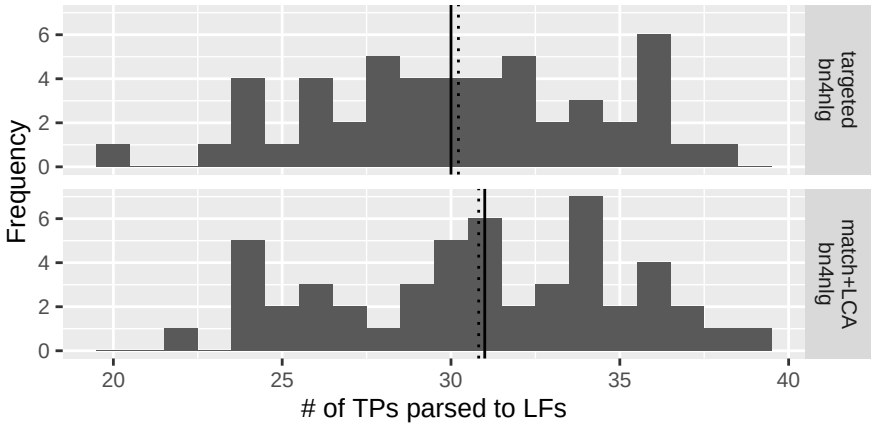


Figure 39: Histogram of parsing coverage for the NovelContrast dataset. The dotted line shows the mean, the solid line shows the median. Each of the 50 datasets generated contained 100 TPs to be parsed.

set of TPs to sample from, we use the procedure described above to generate a new test set consisting of 500 contrastive TPs. Applying our model's rules, we are able to generate LFs for 163 of the TPs using the targeted model and 173 using the match+LCA model. For these LFs we are able to generate 156 and 171 texts, respectively.

9.7.2 *Experimental setup*

We randomly sampled 25 text plans for which we were able to generate LFs and texts for both the match+LCA and targeted `bn4nlg` models. We then used the same scripts to prepare corresponding MRs for TGen and generated 25 texts using the same baseline as before.

In order to keep the experiment as similar to the previous iteration as possible, we included the 25 corpus texts used as control items in the first experiment. This gives us item lists of the same length as in the original experiment and helps us to anchor the current results compared to the earlier results.

We recruited subjects using Prolific Academic and used the same evaluation interface. Each subject took a mean of 35:49 minutes (stddev 10:22) to evaluate 20 texts. Subject ages ranged from 18 to 43 (median 28, mean 29, stddev 7.4) and were each paid 3 GBP for participating.

9.7.3 *Results*

9.7.3.1 *Semantic Fidelity*

Table 16 shows a dramatic improvement in semantic fidelity for `bn4nlg` versus TGen. Since we intentionally did not change the survey format from the previous experiment, we again see that subjects appear

System	Miss.	1	2	3	4	Added
Corpus	12	10	1	0	0	17
TGen	117	52	19	5	3	23
MATCH+LCA	69	45	10	1	0	29
TARGETED	77	51	13	0	0	25

Table 16: Semantic fidelity in the NovelContrast experiment. # of facts dropped by each system (out of 450), and # of instances (out of 125 possible) where a text was missing 1, 2, 3, or 4 facts. No texts dropped more than 4 facts. The last column is the number of instances where a text was marked as having ≥ 1 inserted facts.

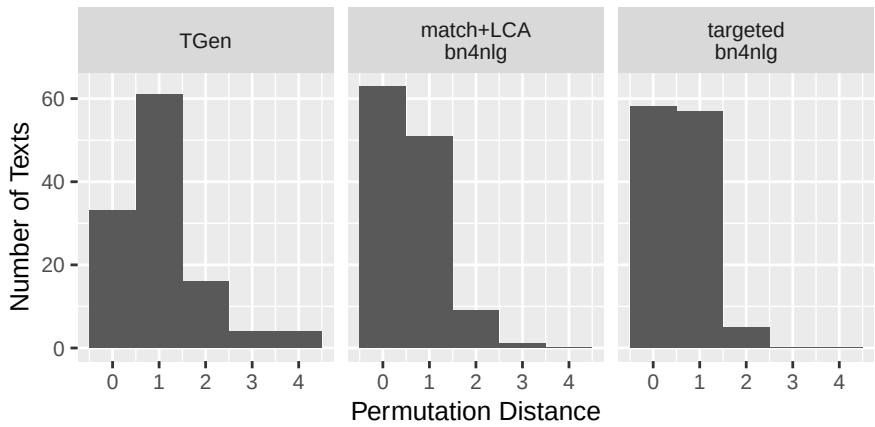


Figure 40: Frequency of different permutation distances for each system in the NovelContrast experiment.

to have interpreted the prompt, 'Does the text include any extra details?', more liberally than we originally intended. The three systems all performed similarly to each other in this regard.

Puzzlingly, four subjects also report the original corpus texts as 'missing' some content, despite all the content being present. These four subjects all provided thoughtful engagement with the 'suggestion' task, suggesting that this was not likely to be accidental.

Looking to information order, Figure 40 shows that both `bn4nlg` systems produce more texts in the correct order than TGen does.

9.7.3.2 *Fluency*

Figure 41 shows the distribution of fluency scores received for each of the systems in this study, as well as for our (not directly comparable) texts from the original corpus. We see that the scores for the original corpus look similar to the scores they received in the first study and that the three systems being compared in this study do not appear to have significantly different distributions.

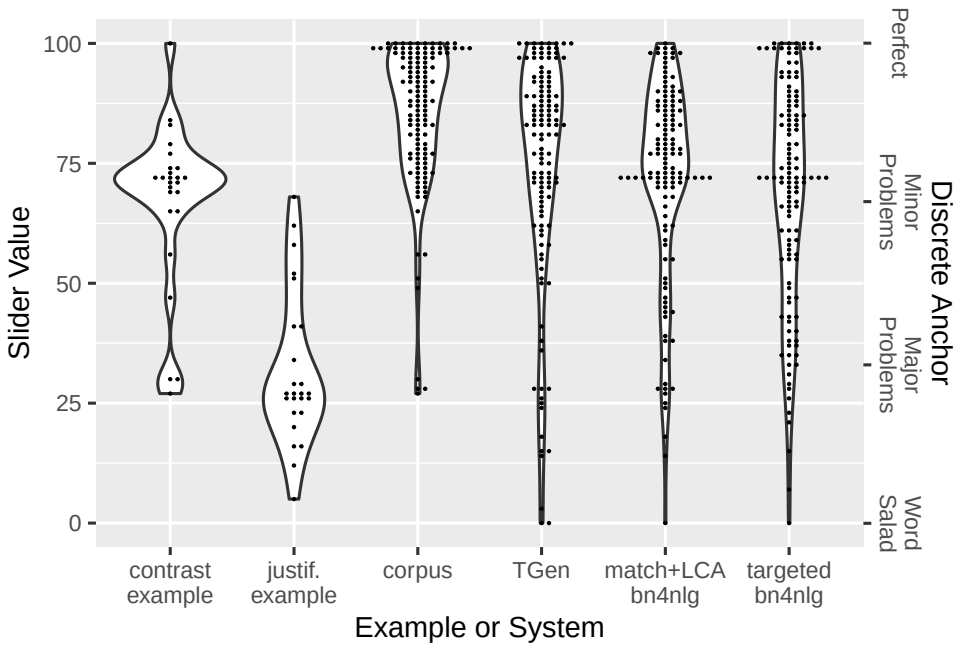


Figure 41: Ratings for instructional texts and each of the systems evaluated for the NovelContrast experiment

We use a post-hoc Wilcoxon signed-rank test to compare each of these three systems to the others, using 3-way Bonferroni correction. None of the comparisons meets the $\alpha = 0.0167$ threshold for significance. The means are 69.3 for the TARGETED `bn4nlg` system, 71.8 for the MATCH+LCA `bn4nlg` system, and 74.3 for TGen.

9.7.3.3 *Discourse*

Figure 42 highlights the responses of participants with respect to the expression of CONTRAST in the texts generated by each system. We see that there does not appear to be a substantial difference between the MATCH+LCA `bn4nlg` system and TGen, although they both do a bit better than the TARGETED `bn4nlg` system.

9.7.4 *Discussion*

Our NovelContrast text plans allowed us to evaluate `bn4nlg` in an environment less similar to the original SRC and while answering questions about how well the system learns rules for expressing discourse relations. We again found no subtantial difference in fluency scores and an improvement in semantic fidelity. This system does not appear to produce substantially better expression of the CONTRAST

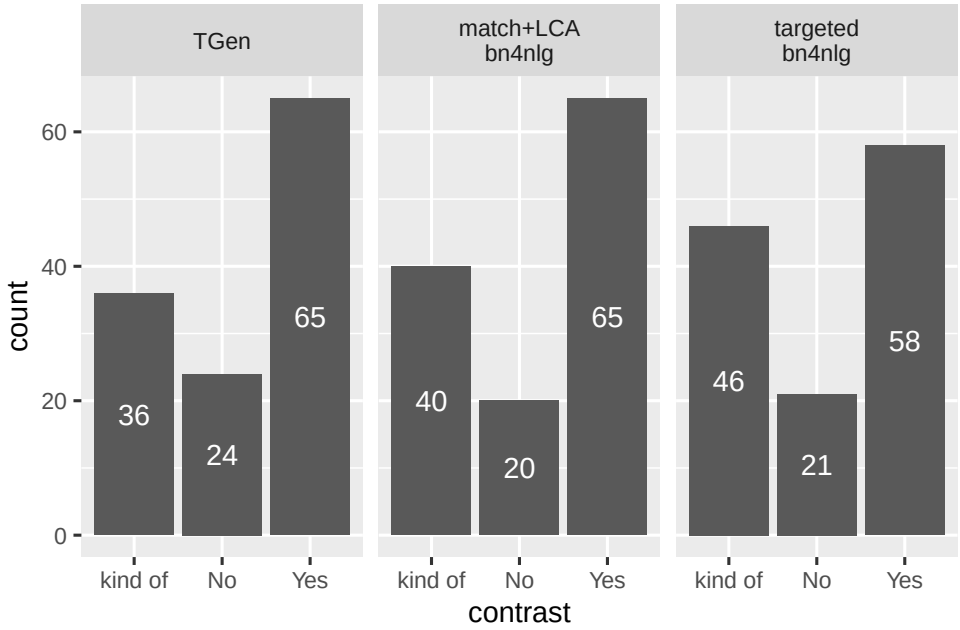


Figure 42: Boxplot of ratings for instructional texts and each of the systems evaluated.

and JUSTIFICATION discourse relations, despite the fact that TGEN's ability to express any discourse relation is entirely incidental.[13]

However, the lack of differences with respect to the expression of discourse relations may be due to the structure of the survey. In the instructions, the contrast relation is introduced by saying:

> For some texts you will be asked about the high-level structure of the text. In this example, we are being asked about whether the text makes a comparison. This text is a bit ambiguous, because it tells us about two different restaurants. Maybe it is doing that in order to implicitly compare them. Read the text and decide for yourself what you think the best answer is for this example.

This approach relies on the participants' intuitons about what is and is not a comparison, but does not define it clearly (e.g. using any of the RST or other annotation guideline definitions). We chose this approach because (1) the survey instructions and format were already rather substantial and we were concerned about pushing the subjects' attention too far and (2) we wanted to assess the quality of judgements based on these kinds of intuitions.

For the justification relation we said:

13 TGEN does not receive CONTRAST or JUSTIFICATION relations as part of the input; any instances of specific discourse connectives related to these relations are due to their presence in the training data.

> Some of our texts are meant to *justify* a claim. This text was supposed to explain *why* "Amy's Bread" is the best restaurant, but it doesn't really try to justify this claim, so we should answer "no" to the next question. Other versions of this text might express the justification explicitly, saying "Amy's Bread is the best restaurant, because it has good food quality..." or "Since it has good food quality and is in Midtown, Amy's Bread has the best overall quality." When you're answering this question, just ask yourself, "Did the system give a reason for one of its claims?"

While this text also encourages participants to rely on their intuitions, it also introduces discourse connectives ('because' and 'since') for explicit expressions of the justification relation. Because discourse annotation is difficult (Scholman & Demberg, 2017) and subjects completed these questions in the context of a long and challenging task, we argue that further study is needed in order to compare the suitability of `bn4nlg` for expressing CONTRAST and JUSTIFICATION.

9.8 EVALUATING ON THE EXTENDED SRC

The SRC dataset provided insight into some basic properties of `bn4nlg`. Because a limited set of hand-crafted NLG rules originally produced the texts in this corpus, the texts are relatively constrained, so our system sees reasonably uniform lexical and syntactic choices. This makes it easy to learn a set of sentence planning rules for generation. However, in order for this approach to be useful in practice, it needs to be able to cope with noisier data like that collected from human participants in the construction of the ESRC (cf. Section 7.4).

9.8.1 *Experimental setup*

In the analyses that follow, we look at the same dev and test inputs as in the SRC experiments. However, for training on the ESRC we incorporate additional preprocessing to delexicalize proper nouns (i.e. restaurant names & neighborhoods) and prices. Since these categories are heavily constrained (e.g. restaurant names must generally be quoted verbatim), we can reliably delexicalize most instances of these categories without removing desired variability from the corpus. Delexicalization increases the overlap between different texts and MRs which should improve the model's ability to find patterns in these more varied texts.

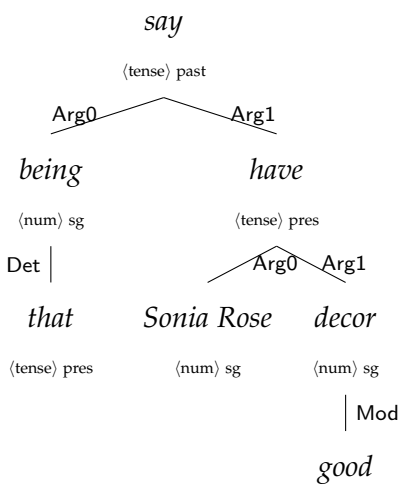


Figure 43: An LF representing the sentence 'That being said, Sonia Rose has good decor'.

9.8.2 *OpenCCG Parsing Errors*

In preparing the data for these experiments, we already encounter one major obstacle to leveraging existing parser-realizers for this task: genre mismatch. OpenCCG's CCGbank grammar is based on edited newspaper English, while the texts elicited for the ESRC are explicitly casual and intended to address familiar interlocutors. This is a problem because any problems with the parses are propagated through `bn4nlg` to the resulting rules, which may not be able to produce valid realizations when fed back into OpenCCG at generation time.

Genre mismatch produces errors above and beyond those addressed by standardizing punctuation and normalizing spelling & capitalization. For example, newspaper texts often include reported speech, such as *The veterinarian said, "Miette most likely ate a lizard and tripped so hard that she lost control of her body from the neck down."*.[14] Comparatively, constructions such as 'that being said', which evoke the concessive relation relative to preceding discourse, are less common in this genre. The result is that the parser interprets 'that being said' as a phrase introducing reported speech: what follows the comma is what a particular (*that*) individual (*being*) said. Figure 43 shows what such a parse looks like in our domain.

9.8.3 *Human Evaluation*

We omit automated metrics from this evaluation, moving on directly to human evaluations to assess the impact of working with more varied data.

14 Paraphrased from Lockwood (2021).

System	Miss.	1	2	3	4	Added
Corpus	0	0	0	0	0	16
TGen	44	35	3	1	0	15
esrc-trained	101	36	8	7	7	28
orig-src-trained	12	10	1	0	0	22

Table 17: Semantic fidelity in the experiment where models are trained on the ESRC. # of facts dropped by each system (out of 408), and # of instances (out of 125 possible) where a text was missing 1, 2, or 3 facts. No texts dropped more than 4 facts. The last column is the number of instances where a text was marked as having ≥ 1 inserted facts.

For our human evaluation we again use TGen as a baseline (cf. Section 9.6.1), trained on the ESRC dataset with the same delexicalization pre-processing as `bn4nlg`. We use the MATCH+LCA `bn4nlg` model and include texts from the SRC test set as well as texts generated by MATCH+LCA `bn4nlg` trained on the SRC for comparison.

9.8.3.1 *Recruiting participants*

We again recruited subjects using Prolific Academic. Participants in this study took a mean of 42:45 minutes (std dev 17:03) to evaluate 20 texts, 5 from each of the four conditions described above. They ranged in age from 18 to 69 (median 25, mean 31, stddev 14) from a variety of socioeconomic backgrounds and were also paid 3 GBP for their participation initially. Due to the higher average completion times, we raised pay after the fact to ensure that participants earned at least 6 GBP/hr on average.[15]

9.8.3.2 *Results: Semantic Fidelity*

Table 17 shows the same kind of semantic fidelity analsyis presented in the preceding sections. For this selection of textplans from the test-set, participants found that the corpus texts indeed included all of the intended information. Again, however, participants have interpreted the 'extra details' question more liberally than intended.

Though both TGen and `bn4nlg` suffer in terms of semantic fidelity when trained on the ESRC, we again find good semantic fidelity scores for `bn4nlg` trained on the SRC.

Figure 44 shows that subjects mostly agree that the corpus is in the correct order, which we know to be objectively true. Both `bn4nlg` models appear to do a better job of preserving the intended order of mention for facts than TGen, though the model trained on SRC does better than ESRC.

15 This was facilitated by new features added in Prolific which make it easy to increase pay after the fact.

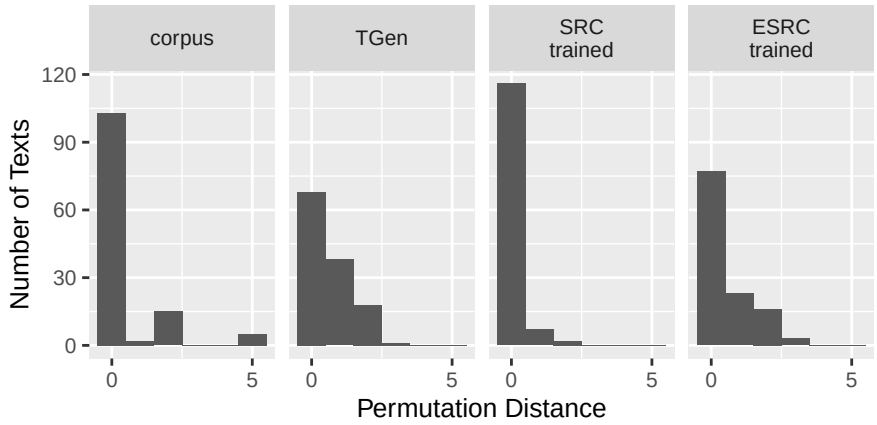


Figure 44: Frequency of different permutation distances for each system in the experiment where models are trained on the ESRC.

9.8.3.3 *Results: Fluency*

The first two columns of Figure 45 shows that participants used the scale similarly to the original evaluation (Sec. 9.6.6) for our training items. The remaining columns show ratings for the original corpus texts, one model trained on the SRC, and two models trained on the ESRC. The 'corpus' column shows that participants assigned somewhat lower ratings to this set of corpus texts compared to the original set of corpus texts (mean score of 80.8 vs. the earlier 89.2). Our original best performing model (MATCH+LCA `bn4nlg`) trained on the SRC (SRC trained) scores quite similarly to the corpus for these MRs (79.4 versus 80.8; difference not significant).

However, both TGEN and this MATCH+LCA `bn4nlg` model trained on the ESRC score substantially worse, with means of 72.5 and 56.6, respectively. All other pair-wise comparisons among these means are significant by a post-hoc Wilcoxon signed rank test with Bonferroni correction, except for the one previously mentioned and the comparison between TGEN trained on the ESRC and and `bn4nlg` trained on the SRC (72.5 vs. 79.4).

Figure 46 shows the range of scores for individual items and indicates that the much lower score for `bn4nlg` versus TGEN overall is likely due to a few items in particular that `bn4nlg` failed to express well, which TGEN managed to express well (the items near the middle of the plot for ESRC trained and TGEN in the figure).

9.8.3.4 *Results: Discourse*

The textplans sampled from the testset for this evaluation contained a better balance of CONTRAST and JUSTIFICATION relations, with 18 textplans expressing the former and 7 expressing the latter. Figure 47 shows the results for CONTRAST. Participants stated that the text (kind

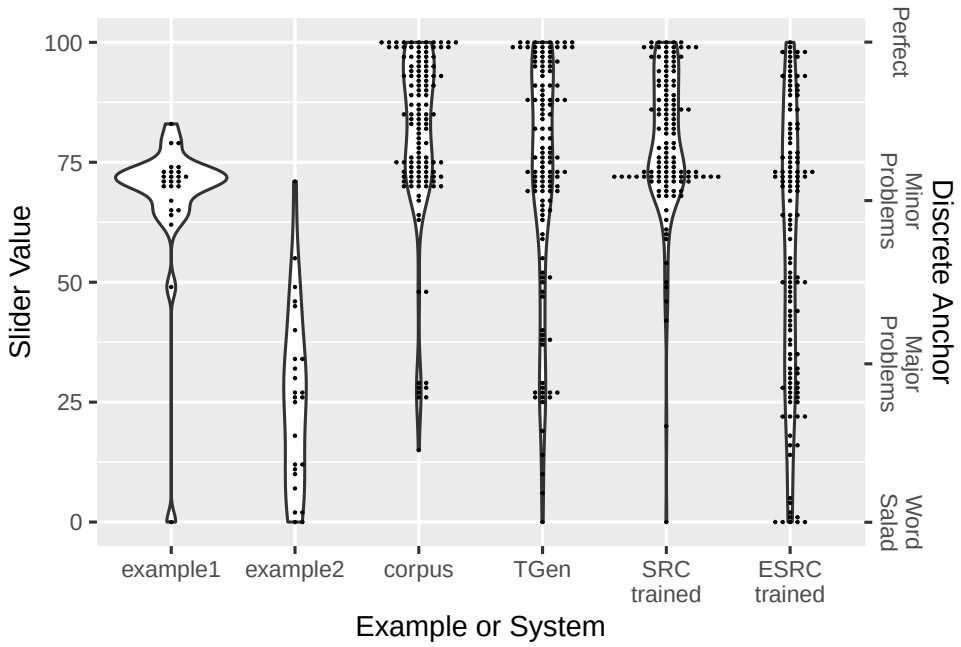


Figure 45: Ratings for instructional texts and each of the systems evaluated in the experiment where models are trained on the ESRC.

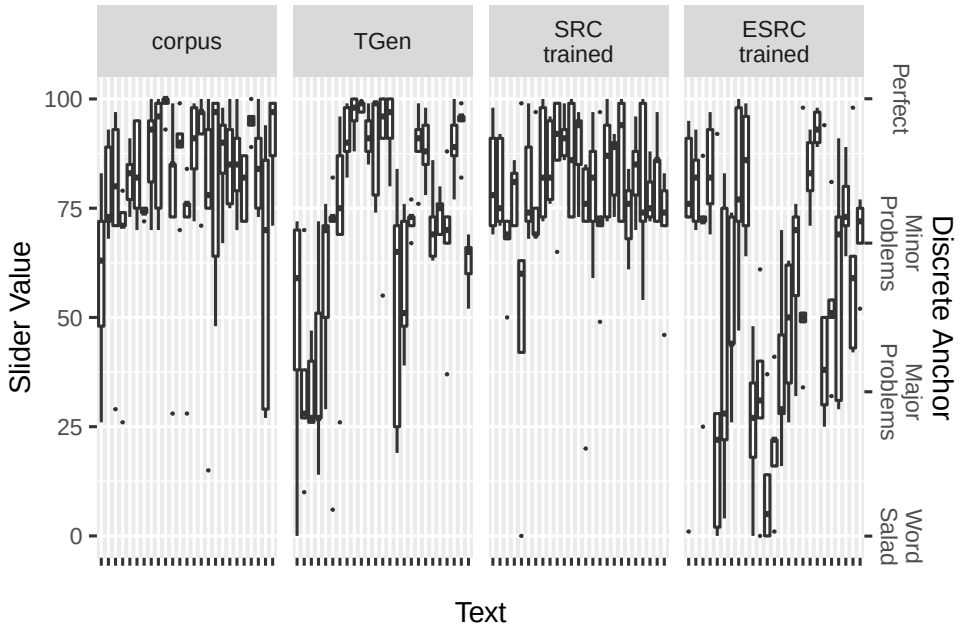


Figure 46: Boxplot of ratings for each item rated in the experiment where models are trained on the ESRC, split out by system.

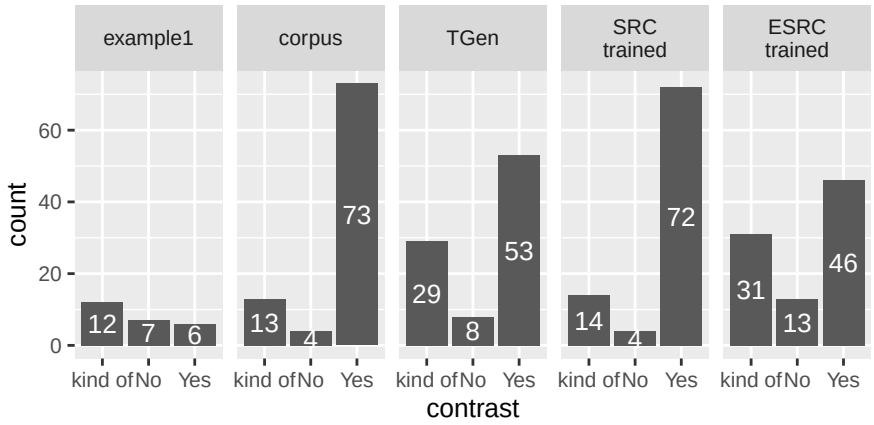


Figure 47: Histograms of responses for texts which were supposed to express CONTRAST in the experiment where models are trained on the ESRC.

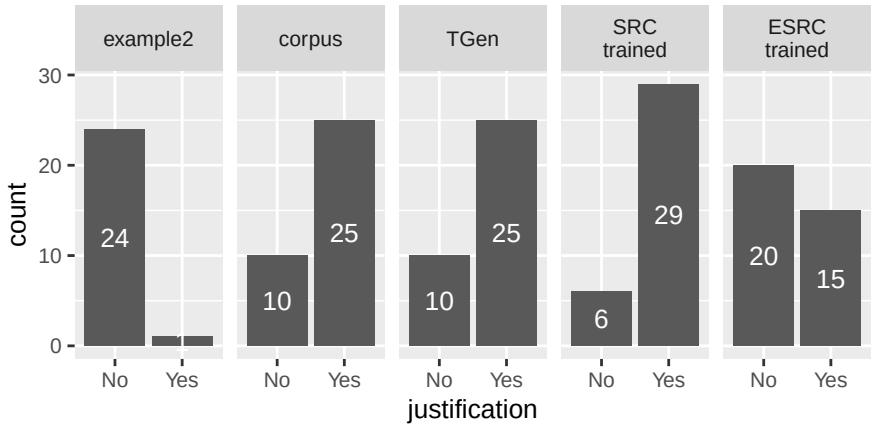


Figure 48: Histograms of responses for texts which were supposed to express JUSTIFICATION in the experiment where models are trained on the ESRC.

of) expressed contrast 73 (13) times in the original corpus texts, out of 90 ratings. A similar proportion of ratings indicated that `bn4nlg` trained on SRC expressed contrast (72 stating yes; 14 stating 'kind of'). However, `bn4nlg` trained on ESRC data was the system most likely to fail to express contrast, with 13 failures compared to TGEN's 8.

Figure 48 shows the results for JUSTIFICATION. The difference is more stark for this relation, with the corpus, `bn4nlg` trained on SRC, and TGEN all showing similar frequency of successfully expressing JUSTIFICATION while `bn4nlg` trianed on ESRC data failed more often than it succeeded at expressing the relation.

9.8.3.5 *Discussion*

The results for bn4nlg trained on ESRC are disappointing. While bn4nlg manages to express facts in the correct order more often than TGEN, it omits more than twice as many facts and inserts nearly twice as many not present in the input. Fluency and the expression of discourse relations are also substantially worse.

Based on our findings during the data preparation stage, it is possible that this is in part due to other tools in our pipeline (i.e. the reversible parser-realizer we depend on for surface realization). This highlights one potential risk of embedding structural knowledge in a model: genre mismatch has a greater impact when there are more components which can differ between genres. In contrast, TGEN's reliance on surface form sequence probabilities without underlying linguistic structure seems to make it more robust to changes in genre.

9.9 DISCUSSION & CONCLUSION

In this chapter we have described a hierarchical Bayesian model for sTSGs and evaluated this model comprehensively using both human & automated evaluations and three different datasets. We found that this model was robust to choice of α and method of model weight initialization (i.e. default versus sampling). On the other hand, the choice of random seed, how to handle root nodes when creating sTSG rules, and treatment of predicate level (i.e. assert) nodes in initializing alignments between TPs and LFs could have a large impact on *coverage*, or the ability of a learnt set of rules to parse a given set of text plans.

Human evaluations revealed that bn4nlg outperforms the neural baseline model on the SRC corpus when it comes to semantic fidelity, content ordering, and expressing the JUSTIFICATION relation while maintaining comparable fluency. Moreover, fluency, semantic fidelity, and content ordering were also better than the baseline when evaluated on a dataset consisting of novel textplans containing only the CONTRAST relation. However, in this second experiment bn4nlg did not do a substantially better job of expressing CONTRAST.

Of course, an important goal in developing a ML approach to building NLG systems is to be able to learn from noisy data produced by humans with minimal editing and annotation oversight. Therefore we also explored the performance of bn4nlg when trained on the ESRC. In this setting, performance was substantially worse, preserving its lead over TGEN only with respect to content ordering.

In addition to these findings relating to our approach to ML for NLG, we also saw evidence that our human evaluation could be improved. Participants were apparently unclear as to what was meant by 'extra details' being included in the text relative to the given semantic in-

puts. We also did not give extensive training in how to identify the presence or absence of discourse relations, due to the already lengthy nature of the survey, preferring instead to rely on their implicit understanding of these terms.

We suggest that future work should disentangle these aspects of evaluation to simplify the task for crowdsourced participants, rather than trying to maximize the feedback gained from each participant across all areas of interest.

Part IV

THE NEED FOR VARIATION

Human responses vary depending on their audience and the context in which they speak, and human listeners react differently to these different texts.

10 HUMAN VARIATION IN REFERRING EXPRESSION GENERATION

Our focus on learning to generate texts exhibiting greater linguistic variation is in part motivated by our desire to build *adaptive* dialogue systems. We want to build systems which can adapt to different users and different situations. In collecting our training corpora (Ch. 7), we emphasized different user groups (younger versus older users) and focused on written language.

These data show how humans adapt their language use when given time to edit a text and explicit instructions to do so; however, we are also interested in understanding how speakers change their utterances based on the situation. To this end we developed a dual-tasking experiment wherein speakers' interlocutors had to complete a non-linguistic task while identifying an object referred to by the speaker. In addition to addressing psycholinguistic issues surrounding human language production, the resulting corpus fills a gap in the REG literature, providing a German dataset for cross-linguistic comparisons of referring expression generation algorithms.

The next section provides background information for the experiment. After that, we describe the materials and procedure, along with findings from the experiment. We discuss the implications of these findings for future research with an emphasis on NLG and dialogue systems and describe the corpus of referring expressions created through this work.

10.1 BACKGROUND

This experiment combines a linguistic task with a non-linguistic task. This section begins by introducing referring expression generation as it has been studied in the NLG community, which forms the basis of our linguistic task. We then shift to findings relevant to our secondary task of choice: driving.

10.1.1 *Referring Expression Generation*

Referring Expression Generation (REG) has been studied for decades in the NLG community (Dale, 1989; Dale & Reiter, 1995; Krahmer, Van Erk & Verleg, 2003; van Deemter, 2002). In particular, the community has been interested in the task of generating referring expressions (REs) which can pick out a single object from a set of similar

objects (called distractors).[1] While it is straightforward to define an algorithm for this task given an appropriate semantic representation for the objects in question (e.g., using the Incremental Algorithm of Dale & Reiter (1995)), such algorithms do not in general model human behavior.

With increasing interest in data-driven methods for NLG, the community set out to better understand *human* REG. General purpose corpora do not contain the degree of semantic annotation required for understanding which properties of an object are mentioned in what circumstances, so van Deemter, van der Sluis & Gatt (2006) designed a controlled experiment for constructing a corpus specialized for answering these questions.

Van Deemter et al. built the TUNA corpus[2] by recruiting human subjects to write English descriptions which uniquely identified a target object from an array of similar distractors. These images were drawn from two domains: the FURNITURE domain and the PEOPLE domain. Images in the former domain consisted of pieces of furniture drawn from the Object Databank (tarrlab, 1996) and differed systematically along 3 dimensions aside from object type: color, size, and orientation. More complex was the PEOPLE domain, which featured black and white photographs of male mathematicians.

Because van Deemter et al. had complete control over the images presented to the users and the set of distractors, they were able to design a systematic evaluation of human REG. However, the second domain presented some challenges for this analysis, as subjects were able to find attributes to use in their descriptions which the experimenters did not anticipate and annotate.

While this experiment was the first to systematically explore human REG for this kind of task, it only examined *written* language. Therefore, when Koolen & Krahmer (2010) set out to build a similar dataset for Dutch, they extended the paradigm to also collect spoken language. For their experiment they used the same stimulus domains (FURNITURE and PEOPLE) and included three different conditions: one focusing on written REs, as in (van Deemter, van der Sluis & Gatt, 2006); and two focusing on spoken REs. These two spoken conditions differed with respect to whether the subject's interlocutor was visible or not.

Our experiment built on these previous approaches. Between the difficulties encountered in previous analyses of the PEOPLE domain and distortion of these images in our experimental setting, we chose to focus only on the FURNITURE domain. Where Koolen & Krahmer (2010) used a confederate as the listener for the experimental subjects, our experiment instead uses subjects for both roles. This ensures that

1 See, however, Krahmer & Van Deemter (2012) for a general survey of the REG for NLG literature, including issues with definining the task so narrowly.

2 TUNA comes from the project name, 'Towards a UNified Algorithm for the generation of referring expressions'.

the speakers' interlocutors are always provide natural reactions to their utterances and avoids any effects related to boredom or difficulty in role-playing on the part of the experimenters.

While the next sections will focus on the psycholinguistic implications of our experiment, it is worth highlighting the significance of the corpus resulting from our experiment. Previous corpora of referring expressions in German either used a virtual environment (Gargett et al., 2010, GIVE-2) or involved an instruction-giving task (Zarrieß et al., 2016, PentoRef), resulting in corpora whose referents are too different from the TUNA corpora for cross-linguistic comparisons of the adequacy of REG algorithms. Our corpus, on the other hand, can be more readily compared with other TUNA corpora. In addition to the English and Dutch corpora we have already described, other researchers have recently contributed comparable corpora for Arabic (Khan, 2016) and Mandarin (van Deemter et al., 2017).

10.1.2 *Language use in the car*

For our non-linguistic task, we chose to use driving in a simulated automobile. Unlike *n*-back and other secondary tasks commonly used in multi-tasking research, driving is an activity that many licensed individuals participate in regularly. Moreover, it is extremely common for drivers to speak with passengers while driving.

Previous research from the automotive literature sought to clarify the impact of conversation on driving safety. In particular, Crundall et al. (2005) found that both drivers and passengers reduce the number of utterances they make as driving situations become more difficult and Drews, Pasupathi & Strayer (2008) found that they tend to use shorter words in more difficult driving conditions.

These findings laid the groundwork for Demberg et al. (2013) to look closely at linguistic and driving dual-tasking. For this work they used a more tightly controlled driving task, called Continuous Tracking and Reaction (ConTRe). ConTRe was developed by Mahr et al. (2012) to examine the effect of 'distractions' on driving performance. Based on the standard lane-change task, in which drivers have to steer a simulated car safely between different lanes when instructed, the ConTRe task requires drivers to steer the car continuously and react to instructions to brake and accelerate. This provides a fine level of control to the researcher investigating multi-tasking with driving as one of the tasks.

This level of control allowed Demberg et al. (2013) to look at the impact of subject- and object-relative clauses on driving performance and to look at the combined impact of these tasks on cognitive load using the Index of Cognitive Activity (ICA), an eye-tracking measure (Marshall, 2002). Subjects had to perform the ConTRe steering task continuously while listening to sentences containing either subject-

or object- relative clauses and then answer comprehension questions about these sentences. The fact that ConTRe is continuous was essential to the comparisons; other driving tasks which involve discrete reactions to stimuli (e.g. changing lanes abruptly) cannot show when a subject is impaired momentarily unless the impairment coincides with the presentation of such a stimulus. Ultimately, Demberg et al. found that steering deviation increased (i.e. driving performance declined) and ICA increased as subjects were processing the object-relative clauses. This is in line with our expectations about subject- versus object-relative clauses and provides validation of the approach to examining the interaction between driving and linguistic tasks in this paradigm.

Our study therefore builds on this paradigm, using a more immersive simulator (see Sec. 10.2.1) for an increase in the naturalness of the driving task. While we did collect comprehension measures for the drivers in our study as well, the emphasis in our work is rather on language production: we examine the utterances produced by a speaker co-present with the driver, in a paradigm explained in Sec. 10.3.2.

10.2 EXPERIMENTAL ENVIRONMENT

Figure 49 depicts our experimental set-up schematically, while a photograph of the set-up can be found in Figure 50. The rest of this section explains the driving simulator and other experimental equipment.

10.2.1 *The Driving Simulator*

Our driving simulator consists of the dashboard and seats from a SMART car mounted in an aluminum frame.The viewing area consists of three panels, one directly in front of the car interior with the other two offset at 135 degree angles to the central panel. Each panel has its own projector, all of which were connected directly to the driving simulator PC at the time of this experiment.

For simulation software we used a modified version of OpenDS 3.0[3], the open source driving simulator software developed by the German Research Center for Artificial Intelligence (DFKI). We modified the basic ConTRe task (described above in Sec. 10.1.2) to present an image centered above the road and modified OpenDS to emit a beeping noise to warn drivers if they strayed too far from the task objectives. We also removed prompts to brake or accelerate to slightly simplify the task and use only the continuous performance measures. An additional script received TTL signals over a serial connection to trigger events in OpenDS, as described in Sec. 10.2.3.

3 https://www.opends.eu

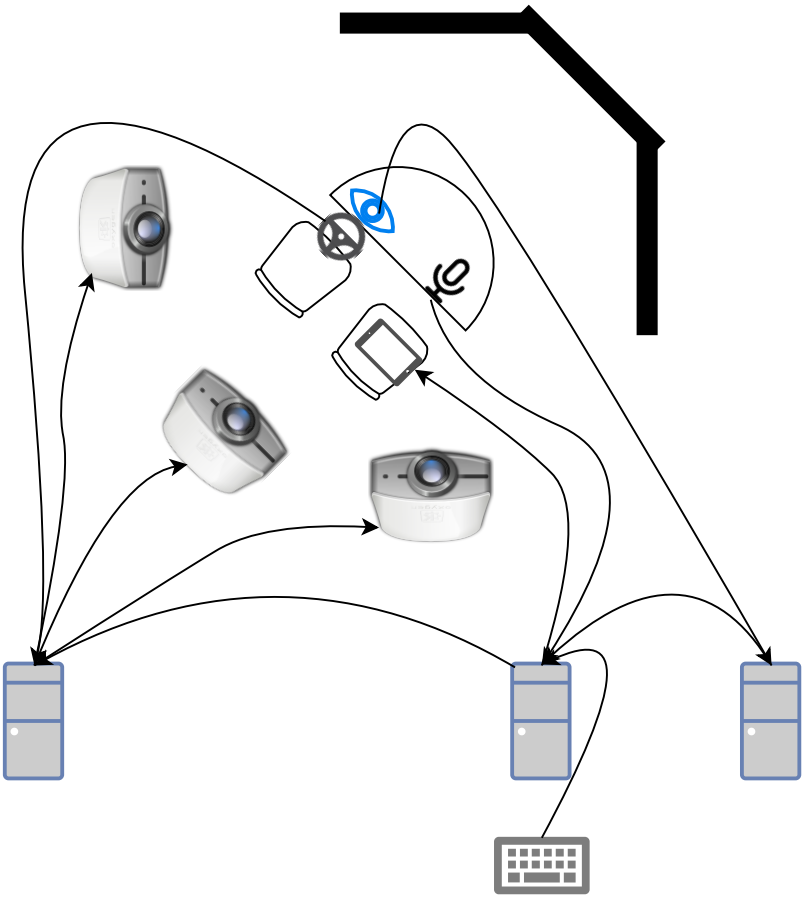


Figure 49: Schematic of experimental set up, showing the driver and passenger seats of the driving simulator, with the dashboard represented as a semicircle. Three projectors display the road and its surroundings on the three-panel screen. The leftmost computer runs the driving simulator, receiving signals from the steering wheel and the central computer. The central computer runs the Experiment Builder software and collects audio and keyboard input. This computer also displays output on the iPad (pictured on the passenger seat). The rightmost computer runs the eye-tracking software based on settings from the central computer.

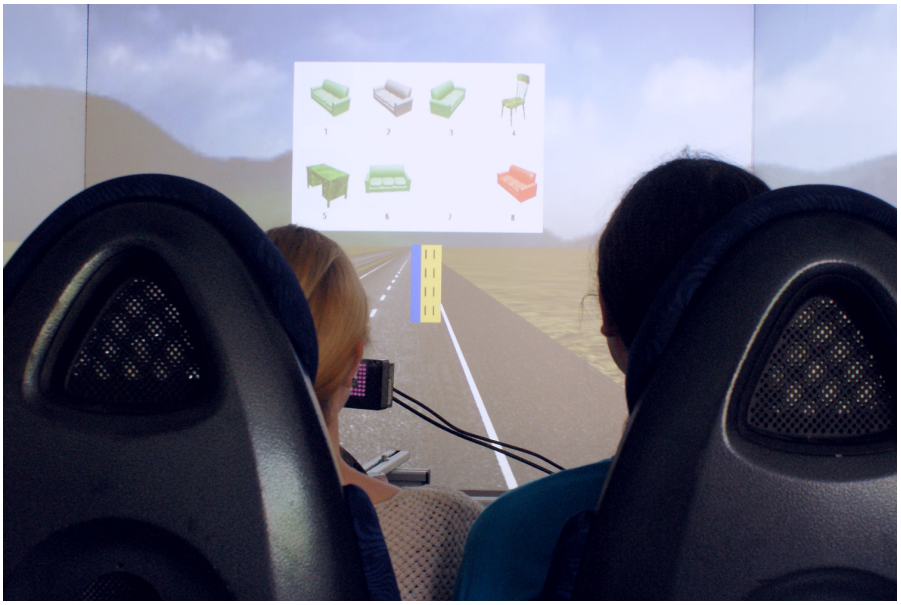


Figure 50: A listener-driver (left) and speaker-passenger (right) seated in the driving simulator. The yellow bar on the road moves left and right, while the driver must steer to keep the blue bar centered in the yellow bar to the best of her ability. Above the road is an array of images, one of which will be described by the passenger. The LEDs visible in front of the driver are the infrared LEDs for the eye-tracker. Not pictured are the iPad used by the passenger to receive instructions or the microphone (mounted in front of the passenger).

10.2.2 *The Eye-Tracker*

In order to assess the cognitive load of the driver and collect gaze information to know where on screen the driver is looking, we used an Eyelink 1000 Plus eye-tracker mounted behind the steering wheel. We situated the eye-tracker such that it did not interfere with the driver's view of the road or the stimuli presented on screen. The eye-tracker sampled both eyes at a rate of 250 Hz.

10.2.3 *Experiment Builder*

To control the experiment we used the Experiment Builder (EB) software from SR Research[4]. Experiment Builder provides a visual interface for designing psychological and psycholinguistic experiments.

EB coordinated with the eye-tracker to begin and end recording during experimental blocks and synchronize this data with the collected audio. The software also sent TTL signals over a serial-port connection to the PC running OpenDS in order to synchronize driving data and prompt the simulator software to display stimuli to the drivers.

We also used Experiment Builder to display prompts to the speaker-passenger on an iPad connected using the Duet Display software[5], collect audio, and record the driver's responses as input by the experimental staff running the experiment.

10.3 MATERIALS AND METHODS

Now that you understand the relevant background and the environment in which our experiments took place, we detail the stimuli we created and our experimental protocol.

10.3.1 *Materials*

We designed our stimuli to be similar to those used in the TUNA and D-TUNA corpora described in Sec. 10.1.1. These corpora were designed to systematically explore human REG to provide a basis for understanding and designing REG for NLG systems. These studies presented subjects with arrays of images and identified one or more images from the array as the target of a referring expression to be written or spoken by the subject.

In our study, we re-use the FURNITURE images, but present them 7 at a time in a 2×4 grid (with grid positions numbered 1-8).The speaker-passenger received a prompt on their secondary display (cf.

4 https://www.sr-research.com
5 https://www.duetdisplay.com

Sec. 10.2.3) telling them which image to describe based on its number. They then had to refer to this image in such a way that the driver could uniquely identify it from among the distractors in the array.

The furniture images come from the Object Databank (tarrlab, 1996). The chosen images consist of four different objects (a chair, a sofa, a desk, and a fan) presented in four different colors (blue, red, green, and grey), three different orientations (front-, left-, and right-facing)[6], and two different sizes (large or small).

Each scene required a particular number of modifiers to be mentioned in order to uniquely identify the target image. The number of required modifiers is referred to as the *minimal description (MD) length*. Our target stimuli were the image arrays requiring subjects to mention either one or two properties of the target image, but we also included filler stimuli which required either no modifiers (i.e. referring to only the object type was sufficient to identify it) or all three possible modifiers to be mentioned. Each image appeared as a target at most once in the experiment.

minimal description (MD) length

For example, the target in the stimulus array pictured in Figure 50 is image #1, which requires that two modifiers be included, namely the orientation of the object (facing left) and the color of the object (green). One acceptable description of this object is 'Das grüne Sofa, das nach links zeigt'[7]. The use of any additional modifiers beyond those required results in overspecification, while the absence of any required modifier is referred to as underspecification.

Since subjects traded roles in the course of the experiment, playing both the role of the listener-driver and the speaker-passenger, we created two lists of stimuli. Each list contained 60 items, subdivided into two blocks of 30 items. Most of the items in each list were targets requiring the mention of either one modifier (in 18 trials) or two modifiers (in 26 trials). We also created 4 practice trials to add to each list, one for each length of minimal description, for presentation during the training blocks of the experiment.

10.3.2 *Methods*

We recruited 25 pairs of students at Saarland University, with average age 23.4 (std. dev. 3.9). Twent-nine participants were women and the rest were men. All received 10 euros for their participation, and the experiment lasted about 1.5 hours. One pair of participants failed to swap roles halfway through the experiment and were therefore excluded from the analyses reported in Sec. 10.4. All subjects gave written consent, and we anonymized all of the data collected.

6 The original dataset also included backward-facing images, but we felt these were too confusable with the front-facing images once projected on our display.

7 'The green sofa facing left.'

When each pair of participants arrived, they were assigned to their initial roles randomly (either speaker-passenger or listener-driver) and seated in their respective seats in the driving simulator. The speaker-passenger then received an iPad which would display the number corresponding to the target image in each trial. Their job was to describe this target image so that the driver could identify it from among the distractors without referring to it by number or describing its position in the array. Every ten trials the speaker-passenger answered a question on the iPad designed to prompt them to think about the listener-driver's cognitive state[8], however, these responses were not recorded.

In each trial the listener-driver then had to identify the target image by stating the corresponding number aloud, which the experimenter then recorded. If the listener-driver failed to respond within 15 seconds, the next trial began automatically.

In addition to this linguistic task, the listener-driver had to steer the car in the driving simulator. However, because the vehicle maintained a constant speed in our experiment, they did not have to accelerate or decelerate. There were two different difficulty settings for this task In the easy setting the driver could simply keep the steering wheel centered and focus on the linguistic task. In the difficult setting the driver had to steer the car in our modified version of the ConTRe task (described in Sec. 10.1.2).

The experiment began with a practice session for subjects to familiarize themselves with their current roles. The practice session included one block of driving only, one block of 4 trials focusing only on the linguistic task, and one block of 4 trials including both the driving and the linguistic tasks. The experimenter calibrated the eye-tracker after this practice session and re-calibrated the eye-tracker between the two subsequent blocks.

The two experimental blocks consisted of thirty trials each, one block including only the easy driving condition and the other including only the difficult condition, in random order. This was randomized across the study so that half of the participants saw the easy driving condition first and half saw the difficult condition first. All but one pair of participants received the blocks in the same order between the two halfs of the experiment. At the end of the two blocks, the subjects switched roles and repeated the procedure, including the practice trials.

10.4 MEASURES AND RESULTS

This section lays out the findings of the experiment, while the next section goes into more detail about the resulting corpus and its relevance to natural language generation. These findings are reported

8 'Wie abgelenkt finden Sie den Fahrer jetzt?'

here for their relevance to our interest in variation in human speech production; however, we leave the details of the statistical analsyses for the reader to seek out from Vogels et al. (2020), because they were not conducted by the author of this thesis.

10.4.1 *Referring expression redundancy*

Our referential task allowed us to operationalize redundancy in terms of over- & under-specification. When speaker-passengers mentioned more attributes than necessary to identify the target object, their utterance was over-specified and therefore redundant. When they failed to mention necessary attributes, their descriptions were underspecified.

Our hypothesis was that we would see speakers increase the redundancy of their referring expressions when drivers were under cognitive load, at least when they had previously experienced the driving task themselves. Our analysis, however, found no general effect of driving difficulty on the redundancy of the generated referring expressions. This effect also did not hold among speakers familiar with the driving task.

However, a post-hoc analysis revealed that speakers who were familiar with the driving task did exhibit *some* adaptation. In particular, if the first driving block during which they were tasked with speaking was a difficult driving block, then they exhibited more redundancy throughout *all* of their referring expressions.

That is, where we had predicted that there would be fairly local adaptation at the level of the individual driving blocks, we instead see what might be called 'coarse' adaptation, with subjects choosing a referential strategy in the first driving block and sticking to it throughout the rest of the experiment.

10.4.2 *Description length*

We found that subjects used more words when they were required to mention more attributes in order to uniquely identify a referent, which is not surprising. However, we also found that word durations were overall shorter when subjects had to mention more attributes.

10.4.3 *Speech rate*

There was no significant association between redundancy and speech rate; however, speakers who had previously driven tended to pronounce all modifiers faster in the difficult condition, regardless of the redundancy of their utterance.

10.4.4 *Driver measures*

Steering deviation was higher when utterances were more redundant. There was no significant difference in response onset and comprehension was at ceiling.

10.5 DISCUSSION

10.5.1 *Human adaptation*

In designing our experiment, we accounted for the fact that speaker-passengers might not recognize the difficulty of the driving task unless they had previously performed the task themselves. However, we did not anticipate that subjects with this experience would choose an initial referential strategy during their first block as a speaker and then fail to update their strategy in the second block, when the driving condition changed. Setting aside for the moment the question of whether or not this behavior is optimal for their listener-drivers, this finding suggests a reasonable starting point for adaptive generation. If our goal is to develop NLG systems which exhibit human-like behavior, we do not necessarily need to continuously update the system's predictions of what behavior is optimal, but we can rather focus on choosing a strategy which is *initially* appropriate to the current situation.

Aside from this observation at the level of referring expressions, which may or may not generalize to other aspects of natural language generation, we can make some observations for the benefit of dialogue systems. A dialogue system includes not just natural language generation, but also the speech recognition, speech synthesis, and dialogue management necessary to converse with a human user. The findings in Sections 10.4.2 & 10.4.3 suggest that human speakers attempt to reduce the temporal duration of their utterances by increasing their speech rate as those utterances become necessarily longer due to task constraints. This implies that a dialogue system engaging with a user under cognitive load involving a continuous non-linguistic task is more human-like if it reduces word duration and increases speech rate as the difficulty of the non-linguistic task increases.

These findings, however, are essentially preliminary because they derive from post-hoc analyses and descriptive statistics.They also focus on observations which might make a dialogue system more human-like; however, human speech production is not necessarily optimal for the listener. In the next section we therefore shift our focus to the listener-drivers.

10.5.2 *Human comprehension*

Our experiment focused heavily on speech production, rather than comprehension, but we can still make some observations based on measures related to the listener-driver (Sec. 10.4.4). While listener-drivers performed extremely well on the linguistic task, we observed an increase in steering deviation when utterances were more redundant. This shows that variation with respect to the linguistic form can and does impact driving performance, as found in earlier work (Demberg et al., 2013).

Note, however, the conflict with speaker-passengers' behavior: speaker-passengers who were familiar with the driving task (and encountered a difficult driving block first when speaking) increased the redundancy of their referring expressions. This suggests that the adaptation chosen by human speakers for listeners under cognitive load hindered their performance on the non-linguistic task. In an experiment in a driving simulator, this is not a grave error and may, in part, be the reason our listener-drivers maintained such high accuracy on the linguistic task throughout the experiment. However, when we want to develop in-car dialogue systems, we would prefer that the linguistic task is sacrificed in favor of driving safety.

10.5.3 *G-TUNA corpus*

In addition to these observations from our experiment, we make the corpus of referring expressions available in the same format as the other TUNA corpora (Gatt, van der Sluis & van Deemter, 2008). This format provides information about the target image and distractors, the referring expression produced, and annotations for which properties of the target image were mentioned in the RE. Unlike the D-TUNA corpus, our corpus also includes word and utterance duration information.

We cleaned the corpus by removing instances where subjects did not follow instructions (e.g. describing the wrong image, referring to the target by number, mentioning earlier trials) or there was an interruption or experimental error. We also removed any updates to the referring expression added due to feedback from the listener-driver. This resulted in about 3.9% data loss for a final corpus of 2767 referring expressions. Table 18 provides summary information comparing the resulting corpus to other TUNA corpora.

In this final dataset we found that 45.7% of referring expressions were overspecified, 1.0% were underspecified, and 52.2% were correctly specified, and therefore minimally described. About 1.0% of the referring expressions were incorrect (e.g. referring to color and orientation when color and size needed to be specified). Koolen, Goudbeek & Krahmer (2011) also found a rate of overspecification of about 50%,

	TUNA	D-TUNA	G-TUNA	A-TUNA	M-TUNA
# subjects	45	60	49	35	37
language	English	Dutch	German	Arabic	Mandarin
# trials	20	40	60	7	44
grid size	3×5	3×5	2×4	3×3	$3 \times 5^*$
# targets/grid	1–2	1–2	1	1	1–2
# distractors/grid	6	6	6	6	6
communicative situation	human-computer	no v. invisible v. visible addressee	driver & passenger in driving simulation	human-computer	human-computer
modality	written	written + spoken	spoken	written	written
domains	furniture, people	furniture, people	furniture	furniture	furniture, people
# comparable / total	420 / 2280	400 / 2400	2767 / 2767	245 / 245	407 / 1628

Table 18: Comparison table for five TUNA corpora. The last row lists the number of 'comparable' REs based on domain & cardinality matches, along with the total number of REs in the corpus. Note, however, that only the 400 D-TUNA REs listed here are in the spoken modality as in our experiments; the rest of the corpora are written only. There are an additional 200 textual furniture REs in the D-TUNA corpus as well. *The M-TUNA scenes were not vertically aligned to a grid.

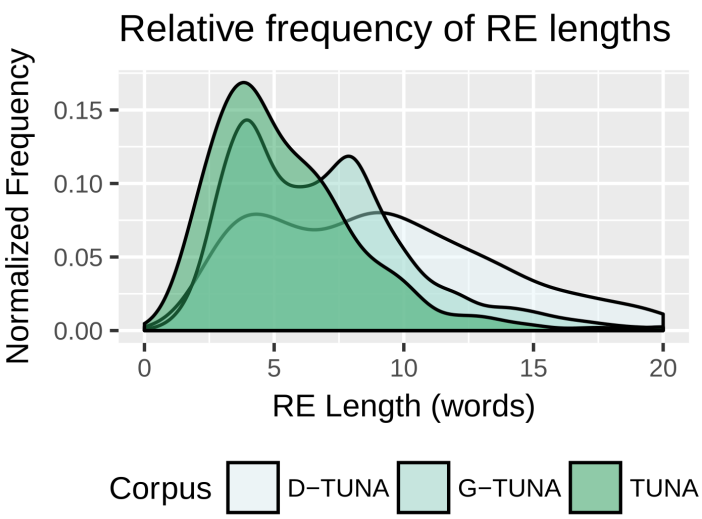


Figure 51: Density plot of RE lengths in the 3 TUNA corpora for comparable REs. The density plot is used so the distribution over different lengths is more easily compared across corpora despite the different numbers of REs in each corpus. From (Howcroft, Vogels & Demberg, 2017).

which is in line with earlier findings that speakers regularly produce overspecified referring expressions.

As shown in Figure 51, the written English descriptions in the TUNA corpus tend to be shorter than the spoken descriptions in the D-TUNA and G-TUNA corpora. We expect the difference between the Dutch and the German corpora to be primarily due to task differences rather than language differences, since the dual-tasking paradigm used in our experiment created time pressure which encouraged speakers to produce shorter utterances.

10.6 CONCLUSION

In this experiment we explored the adaptation of human speakers to listeners who are under cognitive load. While our findings are consistent with the view that speakers adopt some strategies which may be helpful for listeners, further work is needed to determine the extent to which these changes are listener-centric rather than ego-centric.

One important contribution of this work is the G-TUNA corpus of referring expressions in German, which provides a point of comparison for cross-linguistic study in addition to providing a dataset for testing REG algorithms in German. The fact that our corpus contains substantial variation despite the highly controlled nature of this task underscores the importance of variation in natural language. This study, therefore, complements the work presented earlier in this thesis, showing that it is worthwhile to develop resources and algorithms for developing NLG systems which can produce varied outputs.

Part V

OUTLOOK

"But what does it all mean, Basel?" And what remains to be done.

11 DISCUSSION AND CONCLUSIONS

In order for computers to produce natural language texts from non-linguistic information, we need a system for mapping between the two, a system of Natural Language Generation (NLG). We can reduce the difficulty of developing such systems if we leverage Machine Learning (ML) intelligently. While there are many possible approaches to the task, this thesis has argued for one in particular, focusing on sentence planning using synchronous grammars and Bayesian non-parametric methods.

We have developed a representation for sentence plans grounded in Synchronous Tree Substitution Grammar (sTSG) and implemented a model for learning these rules using hierarchical Dirichlet Processes. In order to train our model we collected a novel corpus containing paired texts and discourse structures, and in order to evaluate instantiations of the model we developed a novel interface for human evaluations. We also include a psycholinguistic study which helps to justify our interest in linguistic variation.

11.1 REPRESENTATIONS FOR SENTENCE PLANNING RULES

Sentence planning rules are used to map from a (pseudo-)semantic input representation (a Text Plan (TP)) to a (pseudo-)syntactic output representation (a Logical Form (LF)), performing lexicalization, aggregation, and referring expression generation in the process. In Chapter 3 we described how Synchronous Tree Substitution Grammars (sTSGs) could be used to express such rules, building out the formal representations needed to implement models based on such grammars. We built upon these representations to provide a formal definition of a grammar for dependency trees with attachment similar to (Joshi & Rambow, 2003), which we call Dependency Attachment Grammar (DAG), and extended this grammar to the synchronous setting.

While Tree Substitution Grammar (TSG) and Tree Adjoining Grammar (TAG) have been discussed in the context of NLG for decades, the current work is the first to specifically propose using sTSGs to represent sentence planning rules. This framing facilitates the incorporation of semantic domain and general linguistic knowledge into models for learning to generate. Moreover, we are the first to describe Synchronous Dependency Attachment Grammars (sDAGs), which we argue can be used to go beyond the abilities of sTSGs to represent sentence planning rules in future work.

11.2 MODELS FOR LEARNING SENTENCE PLANNING RULES

Chapter 6 explained the motivation for learning sentence planning rules in particular. By decomposing the NLG task along the lines of the traditional NLG architecture, we are able to simplify the learning task, since the model no longer has to learn every task at once. Moreover, focusing on learning sentence planning rules allows us to leverage existing systems for surface realization, so that our system does not need to learn the morphological, linearization, capitalization, and punctuation quirks of the language it is generating.

Further implementation details related to rule application and surface realization were also explained in that chapter, so that Chapter 9 could instead focus on the particular ML model we implemented, building on prior work in grammar induction for sTSGs (cf. Chapter 5). In Chapter 9 we developed a hierarchical Dirichlet Process to model TSGs for dependency trees before connecting those models together under another Dirichlet Process to model a sTSG for sentence planning rules. We adopted a series of so-called Gibbs operators to perform model updates based on our training data and improve mixing over simpler segmentation models. We explored two approaches to initializing the alignments between TPs and LFs, finding that the simpler approach resulted in better system performance in our evaluations.

To test our model, we performed three sets of experiments. In the first, we applied the model to a test set drawn from a corpus of input-output pairs from an existing NLG system (the SPaRKy Restaurant Corpus (SRC)). This helped to identify which model parameters were and were not relevant to the system's ability to produce outputs for a given set of inputs (i. e. its coverage) and which model parameters resulted in substantial changes in output texts (i. e. output similarity).

When we evaluated the quality of these texts with human judges, we found that the fluency of our model was similar to that of a state-of-the-art neural network baseline while performing substantially better with respect to semantic fidelity, omitting fewer facts given in the input and preserving the intended order of expression.

The test set used for the first human evaluation consisted almost exclusively of JUSTIFICATION relations, however,so we also generated a new set of TPs containing only CONTRAST relations and conducted a second study. This study revealed one weakness of our approach, in that the coverage for our model dropped substantially on this new set of TPs. For those TPs where we were able to generate a text, however, our human evaluation again found comparable fluency compared to our baseline while avoiding omissions and preserving content order.

While this experiment used novel test data, it did not use naturalistic corpus data of the kind a researcher might collect with human participants. Therefore our third experiment focused on our dataset,

the Extended SPaRKy Restaurant Corpus (ESRC), which contains texts with higher and lower degrees of information density.

Neither our baseline nor our model performed especially well on this dataset, with fluency and semantic fidelity dropping dramatically for both, though our model still preserved content order better than the baseline system. However, with this dataset we were able to see the full impact of relying an existing rule-based system with a grammar based on a particular domain: our surface realizer used a grammar from the newspaper domain rather than the informal written domain present in the ESRC. Due to the poor text quality when evaluating on the ESRC, we were not able to further explore the ability of the model to emulate the variation present in the underlying corpus.

The system presented in this thesis is the first system for grammar induction for sentence planning in particular and for synchronous dependency trees in general. Our evaluations highlight weaknesses based on the overall pipeline in which the model is situated while demonstrating that the approach generally does a good job of preserving semantic content and ordering it correctly. This suggests that it is worth exploring alternative implementations in future work (cf. Section 11.5.1).

11.3 NOVEL DATASETS AND LINGUISTIC VARIATION

In Chapter 7 we surveyed current corpora for NLG and the desiderata for training our sentence planning models. Previous corpora contained limited discourse information, representing input not as text plans but rather as a collection of key-value pairs corresponding to facts to be expressed, or contained limited variation, being based on the outputs of existing NLG systems based on a limited set of rules.

Therefore we developed a novel paraphrasing paradigm to crowdsource data collection to train our models on a corpus containing both discourse-structured text plans and varied texts. We found that our experimental manipulation (asking speakers to imagine different audiences for their utterances) was effective in eliciting texts with higher and lower levels of information density. In particular, our participants wrote texts with lower information density when instructed to imagine that they were addressing an elderly relative. We also found that participants often completed the paraphrasing task by re-ordering the information presented in the original text, which we addressed by manually correcting the text plans associated with the original texts to match the texts written by our participants. This resulted in a set of 1344 texts with different levels of information density and gold standard discourse structure annotations. This corpus, which we call the Extended SPaRKy Restaurant Corpus (ESRC) is the first of its kind,

reflecting differences in idea density for short texts directed toward older and younger adults.

In addition to collecting data for our sentence planning task, we collected a novel dataset on the human production of referring expressions (Chapter 10). This corpus is the first German corpus using the same kind of stimuli as earlier corpora for referring expressions in English (van Deemter, van der Sluis & Gatt, 2006) and Dutch (Koolen & Krahmer, 2010), inter alia. Moreover, due to our experimental design, it reflects human behavior when speaking to listeners under cognitive load. The variation present in this corpus reinforces our general claim that NLG systems must be able to produce variation if we want to produce natural texts.

11.4 EVALUATIONS FOR GENERATED TEXT

We examined the state of the art for automatic and human evaluations of NLG systems in Chapter 8 in order to determine the best method for evaluating our system. Our automated metrics focused on raw coverage of possible inputs, identifying which versions of our system were able to produce LFs for the most TPs and how many texts we were able to generate as a result. We also used Bilingual Evaluation Understudy (BLEU) as a text similarity measure to assess the extent to which different parameter settings resulted in different texts.

We developed scripts for rapid assessment of text quality by researchers and a novel crowdsourcing interface for assessment by crowd workers. Our quick comparison script achieved its goal of allowing a researcher to quickly assess the relative quality of different texts, completing comparisons of 100 text pairs in just 20 minutes. More importantly, our evaluation interface for crowd workers used provided a way to collect fine grained scores while still grounding the ratings with descriptive anchors: by using a sliding scale and with textual descriptions along the scale, participants are able to differentiate their scores for texts of similar quality better than with a simple 5-, 6-, or 7-point rating scale without having to guess at what the midpoint of the scale for something as abstract as fluency should be. By collecting continuous data, we were able to use simple parametric statistical tests instead of the more complex models required for ordinal data. The interface also provided feedback on semantic fidelity, though participants struggled with the notion of 'extra details', frequently interpreting this to mean 'did the system include facts you consider irrelevant' as opposed to 'did the system express any facts in addition to those listed above'. The survey also asked about the expression of discourse relations; however, responses to these questions were less informative than they could have been.

While our participants appear to have participated in good faith and made sincere efforts to answer our questions, switching between

fluency judgements, semantic fidelity assessments, and discourse relation perceptions appears to have been difficult for our participants. Therefore we propose that future evaluation surveys should focus on answering only one or two closely related questions at a time; for example, they might focus only on fluency judgements, only on inserted and omitted facts, or only on discourse structure.

11.5 DIRECTIONS FOR FUTURE RESEARCH

11.5.1 *Evaluating the impact of other pipeline components*

In our experiments we observed that our model manages a greater degree of semantic control with comparable fluency to a baseline sequence-to-sequence (seq2seq) model but becomes brittle in the face of some kinds of text variation. While the first Surface Realisation Shared Task (Belz et al., 2011) focused on the same Penn Treebank (PTB) data that the broad coverage English grammar used by OpenCCG is based on, later iterations of the the task include other genres as well (Mille et al., 2018, 2019, 2020) and target languages other than English. Since these tasks use the Universal Dependencies (UD) (de Marneffe et al., 2021) representation as input, future work could use existing off-the-shelf parsers (e. g. Chen & Manning, 2014; Honnibal et al., 2020) to prepare syntactic trees for training and use one of the surface realizers developed for these shared tasks for generation.

A systematic comparison of different combinations of parsers and surface realizers across text genres would make it possible to choose the right combination for a given application, allowing researchers to apply the synchronous grammar induction techniques presented in this thesis with a wider range of texts.

11.5.2 *Embedding more linguistic knowledge in our models*

The statistical model developed in Chapter 9 uses the dependency and node labels present in the training corpora to set priors for the kinds of trees we expect to see in TPs and LFs. However, future work can build upon this to incorporate more information about the semantic or discourse domain, the target language, or the surface realizer to be used downstream.

Semantic and discourse knowledge Input TPs of the kind used in this thesis often have predicates with extremely constrained domains. For example, the Arg0 for all pre-terminal nodes (i. e. our predicates such as CUISINE and PRICE) must always be a restaurant and the Arg1 of FOODQUALITY, DECOR, & SERVICE is always from a closed class of quality terms (e. g. GOOD, SUPERB, etc). A more informative prior would use this knowledge of the semantic domain to ensure that the probabilities for disallowed values are always 0. Sim-

ilarly, future work could restrict the higher levels of TPs to include only discourse relations.

Linguistic knowledge The morphosyntactic LFs used in this thesis only leverage node labels (i. e. words) and arc labels (i. e. dependencies or arguments), but the parses produced by OpenCCG include additional annotations representing tense and number, among other morphosyntactic features. Future work could train the TSG for LFs on a larger dataset (e. g. the entirety of CCGbank) and use this trained model as the prior for LF structure rather than the simpler model based on the training data for the sTSG task as done here. In training on a larger dataset like this, the model could also be extended to explicitly include the kind of tense and number information mentioned above, providing further cues about likely word classes to appear in a particular context.

Surface realizer knowledge Since the LF trees output by a sentence planner must be processed by a surface realizer, it makes sense to use knowledge of that surface realizer in constraining the kinds of elementary trees which are possible. If, for example, an induced elementary tree is incompatible with the grammar used by the surface realizer, regardless of whether or not it is compatible with the grammar of the target language in general, then the prior for such elementary trees could be set to 0 in the model.

Another interesting approach would be to close the loop between grammar application and model fitting. For example, using a validation set of TPs that the system should be able to parse successfully and adjusting the sampling procedure to disprefer rules which result in failures to parse an input TP into an output LF. Similarly, we could downweight rules which produce LFs which the surface realizer fails to generate from.

11.5.3 *Increasing structure in neural models*

This thesis has focused on learning sentence planning rules using Bayesian nonparametric methods; however, it is in principle possible to learn sentence planning rules using neural networks or to use a similar problem decomposition to reduce the difficulty of training a neural NLG model. Castro Ferreira et al. (2019), for example, showed that decomposing the task into content ordering, sentence assignment, and lexicalization resulted in improved performance compared to an end-to-end neural NLG system, and Balakrishnan et al. (2019) used input TPs similar to our own along with a novel constrained decoding approach to improve performance. Future work can build upon these efforts to develop a neural pipeline model with an explicit sentence planning component which takes into account hierarchical discourse structure.

11.5.4 *Improved human evaluations*

Our human evaluations provided the first slider-based human evaluation with multiple qualitative descriptions at anchor points along the scale. The results in Chapter 9 show an interesting pattern of results, where some participants appear to use the scale more continuously while others use it in a more discrete fashion, tending to rate texts very close to one of the anchors. It would be interesting to explore these different strategies in more depth and to assess how data collected in this paradigm compares to traditional rating scales which are strictly discrete or continuous.

While our interface seemed to perform well for assessing content ordering and omissions, there is clear room for improvement when it comes to assessing information insertion (i. e. 'hallucinations') and the presence of discourse connectives. One promising direction for future work would be to decompose the evaluation task and consider using discourse annotation paradigms like those developed in (Scholman, 2018) to allow crowdsourced participants to annotate possible discourse relations in generated texts.

11.6 CONCLUSION

The work presented in this thesis focused on learning sentence planning rules to generate novel texts using synchronous grammars. We defined a formalism for describing these generation rules, collected a novel dataset for training discourse-aware NLG systems, and implemented & evaluated one such system on several datasets. In addition to these practical efforts, we explored human variation in adapting their utterances to listeners under cognitive load.

These efforts indicate that synchronous grammars provide a useful representation for sentence planning rules, that Bayesian nonparametric models can induce such grammars given appropriate training data, and that such learnt models can outperform existing neural network models with respect to semantic fidelity. However, this thesis also opens up several directions for future research into how best to integrate the various challenging tasks involved in natural language generation and how best to evaluate these systems in the future.

BIBLIOGRAPHY

Angeli, Gabor, Percy Liang & Dan Klein (2010). "A Simple Domain-Independent Probabilistic Approach to Generation." In: *EMNLP*. Cambridge, MA: Association for Computational Linguistics, pp. 502–512. URL: http://www.aclweb.org/anthology/D10-1049 (cit. on pp. 93, 95).

Asr, Fatemeh Torabi (2015). "An Information Theoretic Approach to Production and Comprehension of Discourse Markers." PhD thesis (cit. on p. 73).

Balakrishnan, Anusha, Jinfeng Rao, Kartikeya Upasani, Michael White & Rajen Subba (June 2019). "Constrained Decoding for Neural NLG from Compositional Representations in Task-Oriented Dialogue." In: *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*. arXiv: 1906.07220. URL: http://arxiv.org/abs/1906.07220 (visited on 06/20/2019) (cit. on p. 170).

Baldridge, Jason & Geert-Jan M. Kruijff (2003). "Multi-Modal Combinatory Categorial Grammar." In: *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - EACL '03*. Vol. 1. Budapest, Hungary: Association for Computational Linguistics, p. 211. DOI: 10/dcjc4g. URL: http://portal.acm.org/citation.cfm?doid=1067807.1067836 (visited on 07/12/2019) (cit. on p. 65).

Banerjee & Lavie (2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments." In: *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan, USA: Association for Computational Linguistics, pp. 65–72 (cit. on p. 91).

Bangalore, Srinivas, Owen Rambow & Steve Whittaker (2000). "Evaluation Metrics for Generation." In: *Proc. of the 1st International Conference on Natural Language Generation (INLG)*. Mitzpe Ramon, Israel: Association for Computational Linguistics. ISBN: 978-965-90296-0-0. DOI: 10.3115/1118253.1118255. URL: https://www.aclweb.org/anthology/W00-1401/ (visited on 09/16/2018) (cit. on p. 94).

Becker, Tilman (1998). "Fully Lexicalized Head-Driven Syntactic Generation." In: *Proc. of the 9th International Workshop on Natural Language Generation (INLG)*. Niagra-on-the-Lake, Ontario, Canada: Association for Computational Linguistics, p. 10 (cit. on p. 101).

Belz, Anja (2005). "Statistical Generation: Three Methods Compared and Evaluated." In: *Proc. of the 10th European Workshop on Natural Language Generation (ENLG)* (cit. on p. 101).

Belz, Anja (Oct. 2008). "Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models." In: *Natural Language Engineering* 14.4, pp. 431–455. ISSN: 1351-3249, 1469-8110. DOI: 10/dvvmsr. URL: https://www.cambridge.org/core/product/identifier/S1351324907004664/type/journal_article (visited on 04/16/2019) (cit. on p. 101).

Belz, Anja & Eric Kow (June 2011). "Discrete vs. Continuous Rating Scales for Language Evaluation in NLP." In: *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vol. Short Papers. Portland, Oregon, USA: Association for Computational Linguistics, pp. 230–235 (cit. on p. 104).

Belz, Anja & Ehud Reiter (Apr. 2006). "Comparing Automatic and Human Evaluation of NLG Systems." In: *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Trento, Italy: Association for Computational Linguistics, pp. 313–320 (cit. on p. 90).

Belz, Anja, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan & Amanda Stent (2011). "The First Surface Realisation Shared Task: Overview and Evaluation Results." In: *Proc. of the 13th European Workshop on Natural Language Generation (ENLG)*. Nancy, France: Association for Computational Linguistics, pp. 217–226 (cit. on pp. 64, 93, 94, 169).

Bergen, Leon, Edward Gibson & Timothy J. O'Donnell (2015). "A Learnability Analysis of Argument and Modifier Structure." URL: http://ling.auf.net/lingbuzz/002502 (cit. on p. 56).

Blunsom, Phil, Trevor Cohn, Sharon Goldwater & Mark Johnson (2009). "A Note on the Implementation of Hierarchical Dirichlet Processes." In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, pp. 337–340. URL: http://dl.acm.org/citation.cfm?id=1667688 (visited on 08/09/2017) (cit. on pp. 55, 59).

Bod, Rens (1992). "A Computational Model of Language Performance: Data-Oriented Parsing." In: *Proc. of the 14th Conference on Computational Linguistics (COLING)*. Nantes, France: Association for Computational Linguistics, pp. 855–859 (cit. on p. 56).

Bod, Rens (1993). "Using an Annotated Language Corpus as a Virtual Stochastic Grammar." In: *Proc. of the 11th National Conference on AI (AAAI)* (cit. on p. 56).

Buschmeier, Hendrik, Kirsten Bergmann & Stefan Kopp (2009). "An Alignment-Capable Microplanner for Natural Language Generation." In: *Proc. of the 12th European Workshop on Natural Language Generation (ENLG)*. Athens, Greece: Association for Computational Linguistics, pp. 82–89. DOI: 10.3115/1610195.1610207. URL: http://portal.acm.org/citation.cfm?doid=1610195.1610207 (visited on 08/15/2018) (cit. on p. 16).

Cahill, Aoife (Aug. 2009). "Correlating Human and Automatic Evaluation of a German Surface Realiser." In: *Proc. of the ACL-IJCNLP 2009 Conference*. Vol. Short Papers. Suntec, Singapore: Association for Computational Linguistics, pp. 97–100. DOI: 10.3115/1667583.1667615. URL: http://portal.acm.org/citation.cfm?doid=1667583.1667615 (visited on 06/06/2018) (cit. on p. 94).

Callaway, Charles B. & James C. Lester (Aug. 2002). "Narrative Prose Generation." In: *Artificial Intelligence* 139.2, pp. 213–252. ISSN: 00043702. DOI: 10/b7pr3b. URL: https://linkinghub.elsevier.com/retrieve/pii/S0004370202002308 (visited on 05/27/2019) (cit. on pp. 92–95).

Carlson, Lynn, Daniel Marcu & Mary Ellen Okurowski (2001). "Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory." In: *SIGDIAL*. Association for Computational Linguistics (cit. on p. 16).

Carroll, John, Ann Copestake, Dan Flickinger & V. Poznánski (1998). "An Efficient Chart Generator for (Semi-) Lexicalist Grammars." In: *Proceedings of the 7th European Workshop on Natural Language Generation* (cit. on p. 101).

Castro Ferreira, Thiago, Chris van der Lee, Emiel van Miltenburg & Emiel Krahmer (2019). "Neural Data-to-Text Generation: A Comparison between Pipeline and End-to-End Architectures." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 552–562. DOI: 10/gg7ztr. URL: https://www.aclweb.org/anthology/D19-1052 (visited on 10/25/2020) (cit. on pp. 64, 170).

Chen, Danqi & Christopher D Manning (Oct. 2014). "A Fast and Accurate Dependency Parser Using Neural Networks." In: *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 740–750. DOI: 10.3115/v1/D14-1082. URL: https://aclanthology.org/D14-1082/ (cit. on p. 169).

Cohn, Trevor & Phil Blunsom (2009). "A Bayesian Model of Syntax-Directed Tree to String Grammar Induction." In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, pp. 352–361. URL: http://dl.acm.org/citation.cfm?id=1699557 (visited on 08/09/2017) (cit. on p. 59).

Cohn, Trevor & Phil Blunsom (2010). "Blocked Inference in Bayesian Tree Substitution Grammars." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. ISBN: 978-1-61738-808-8 (cit. on p. 55).

Cohn, Trevor, Phil Blunsom & Sharon Goldwater (2010). "Inducing Tree-Substitution Grammars." In: *Journal of Machine Learning Re-*

search 11, pp. 3053–3096. URL: http://www.jmlr.org/papers/volume11/cohn10b/cohn10b.pdf (cit. on pp. 51–57).

Cohn, Trevor, Sharon Goldwater & Phil Blunsom (2009). "Inducing Compact but Accurate Tree-Substitution Grammars." In: *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 548–556. DOI: 10.3115/1620754.1620834 (cit. on p. 55).

Crundall, David, Manpreet Bains, Peter Chapman & Geoffrey Underwood (May 2005). "Regulating Conversation during Driving: A Problem for Mobile Telephones?" In: *Transportation Research Part F: Traffic Psychology and Behaviour* 8.3, pp. 197–211. ISSN: 13698478. DOI: 10.1016/j.trf.2005.01.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S1369847805000057 (cit. on p. 151).

Dale, Robert (1989). "Cooking up Referring Expressions." In: *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 68–75. URL: http://dl.acm.org/citation.cfm?id=981632 (visited on 05/11/2017) (cit. on p. 149).

Dale, Robert & Ehud Reiter (1995). "Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions." In: *Cognitive Science* 18, pp. 233–263. URL: http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog1902_3/full (visited on 05/11/2017) (cit. on pp. 149, 150).

Demberg, Vera & Frank Keller (2008). "Data from Eye-Tracking Corpora as Evidence for Theories of Syntactic Processing Complexity." In: *Cognition* 109.2, pp. 193–210. ISSN: 1873-7838. DOI: 10.1016/j.cognition.2008.07.008. URL: http://www.ncbi.nlm.nih.gov/pubmed/18930455 (cit. on p. 73).

Demberg, Vera, Asad Sayeed, Angela Mahr & Christian Müller (Oct. 2013). "Measuring Linguistically-Induced Cognitive Load during Driving Using the ConTRe Task." In: *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. DOI: 10.1145/2516540.2516546 (cit. on pp. 151, 160).

Doddington, George (2002). "Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics." In: *Proceedings of the Second International Conference on Human Language Technology Research* -. San Diego, California: Association for Computational Linguistics, p. 138. DOI: 10/bgw6z2. URL: http://portal.acm.org/citation.cfm?doid=1289189.1289273 (visited on 05/25/2019) (cit. on p. 91).

Drews, Frank A., Monisha Pasupathi & David L. Strayer (2008). "Passenger and Cell Phone Conversations in Simulated Driving." In: *Journal of Experimental Psychology: Applied* 14.4, pp. 392–400. ISSN: 1076-898X. DOI: 10.1037/a0013119 (cit. on p. 151).

Dušek, Ondřej, David M. Howcroft & Verena Rieser (2019). "Semantic Noise Matters for Neural Natural Language Generation." In: *Proc.*

of the 12th International Conference on Natural Language Generation (INLG). Tokyo, Japan: Association for Computational Linguistics. DOI: 10/ggwzgc (cit. on p. 77).

Dušek, Ondřej & Filip Jurčíček (2015). "Training a Natural Language Generator from Unaligned Data." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Vol. 1, pp. 451–461. URL: https://pdfs.semanticscholar.org/24db/af93c13c0f47e7df8f8a61bdac1e6b30e66e.pdf (visited on 07/21/2017) (cit. on p. 90).

Dušek, Ondřej & Filip Jurčíček (2016). "Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings." In: *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vol. Volume 2: Short Papers. Berlin, Germany: Association for Computational Linguistics, pp. 45–51. DOI: 10.18653/v1/P16-2008. URL: http://aclweb.org/anthology/P16-2008 (visited on 04/24/2018) (cit. on pp. 19, 90, 128).

Dušek, Ondřej, Jekaterina Novikova & Verena Rieser (Jan. 2020). "Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge." In: *Computer Speech & Language* 59, pp. 123–156. DOI: 10.1016/j.csl.2019.06.009. arXiv: 1901.11528. URL: https://www.sciencedirect.com/science/article/pii/S0885230819300919 (cit. on p. 128).

Eisner, Jason (2003). "Learning Non-Isomorphic Tree Mappings for Machine Translation." In: *Proc. of the 41st Annual Meeting on Association for Computational Linguistics (ACL): Short Papers*. Vol. 2. Association for Computational Linguistics, pp. 205–208. URL: http://www.aclweb.org/anthology/P03-2041 (cit. on pp. 25, 30).

Elliott, Desmond & Frank Keller (2014). "Comparing Automatic Evaluation Measures for Image Description." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 452–457. DOI: 10/gfzj2c. URL: http://aclweb.org/anthology/P14-2074 (visited on 04/17/2019) (cit. on pp. 93, 95).

Espinosa, Dominic, Rajakrishnan Rajkumar, Michael White & Shoshana Berleant (2010). "Further Meta-Evaluation of Broad-Coverage Surface Realization." In: *EMNLP*, pp. 564–574. URL: http://www.aclweb.org/anthology/D10-1055 (cit. on pp. 90, 93, 95).

Espinosa, Dominic, Michael White & Dennis N. Mehay (2008). "Hypertagging: Supertagging for Surface Realization with CCG." In: *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 183–191. URL: http://aclweb.org/anthology-new/P/P08/P08-1022.pdf (cit. on p. 90).

Gabriel, Richard P. (1988). "Deliberate Writing." In: *Natural Language Generation Systems*. Symbolic Computation. Springer-Verlag, pp. 1–46. ISBN: 0-387-96691-9 (cit. on p. 101).

Gargett, Andrew, Konstantina Garoufi, Alexander Koller & Kristina Striegnitz (2010). "The GIVE-2 Corpus of Giving Instructions in Virtual Environments." In: *LREC*. URL: https://www.ling.uni-potsdam.de/~koller/papers/give-corpus-10.pdf (visited on 05/12/2017) (cit. on p. 151).

Gatt, Albert & Emiel Krahmer (Jan. 2018). "Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications, and Evaluation." In: *Journal of Artificial Intelligence Research* 61, pp. 65–170 (cit. on pp. 91, 92).

Gatt, Albert, François Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur & Somayajulu Sripada (2009). "From Data to Text in the Neonatal Intensive Care Unit: Using NLG Technology for Decision Support and Information Management." In: *AI Communications*, 153ß186. DOI: 10.3233/aic-2009-0453. URL: https://hal.archives-ouvertes.fr/hal-00953706 (cit. on p. 96).

Gatt, Albert, Ielka van der Sluis & Kees van Deemter (2008). *XML Format Guidelines for the TUNA Corpus*. Tech. rep. Technical report, Computing Science, Univ. of Aberdeen, http://www. csd. abdn. ac. uk/ agatt/home/pubs/tunaFormat. pdf. URL: https://www.abdn.ac.uk/ncs/documents/xml_format_tuna.pdf (visited on 05/12/2017) (cit. on p. 160).

Gkatzia, Dimitra & Saad Mahamood (2015). "A Snapshot of NLG Evaluation Practices 2005 - 2014." In: *Proc. of the 15th European Workshop on Natural Language Generation*. Brighton, England, United Kingdom: Association for Computational Linguistics, pp. 57–60 (cit. on pp. 90, 96).

Goldwater, Sharon, Thomas L. Griffiths & Mark Johnson (2006). "Contextual Dependencies in Unsupervised Word Segmentation." In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL - ACL '06*. Sydney, Australia: Association for Computational Linguistics, pp. 673–680. DOI: 10/fc525h. URL: http://portal.acm.org/citation.cfm?doid=1220175.1220260 (visited on 01/28/2019) (cit. on pp. 47, 53).

Goldwater, Sharon, Thomas L. Griffiths & Mark Johnson (2007). "Distributional Cues to Word Boundaries: Context Is Important." In: *Proceedings of the 31st Annual Boston University Conference on Language Development* (cit. on p. 47).

Gu, Jiantao, Qi Liu & Kyunghyun Cho (2019). "Insertion-Based Decoding with Automatically Inferred Generation Order." TACL Preprint (cit. on p. 90).

Gyawali, Bikash & Claire Gardent (2014). "Surface Realisation from Knowledge-Bases." In: *Proceedings of the 52nd Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore, Maryland: Association for Computational Linguistics, pp. 424–434. DOI: 10.3115/v1/P14-1040. URL: http://www.aclweb.org/anthology/P/P14/P14-1040 (cit. on pp. 93, 95).

Hastie, Helen, Heriberto Cuayahuitl, Nina Dethlefs & Simon Keizer (2016). "Evaluation of NLG in an End-to-End Spoken Dialogue System- Is It Worth It?" In: *Dialogues with Social RobotsEnablements, Analyses, and Evaluation*. Lecture Notes in Electrical Engineering. ISBN: 978-981-10-2584-6. URL: http://www.macs.hw.ac.uk/~hh117/pubs/iwsds2016.pdf (cit. on p. 96).

Hockenmaier, Julia & Mark Steedman (Sept. 2007). "CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank." In: *Computational Linguistics* 33.3, pp. 355–396 (cit. on p. 65).

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem & Adriane Boyd (2020). *spaCy: Industrial-Strength Natural Language Processing in Python*. Zenodo. DOI: 10.5281/zenodo.1212303. URL: https://doi.org/10.5281/zenodo.1212303 (cit. on p. 169).

Howcroft, David M. & Vera Demberg (2017). "Psycholinguistic Models of Sentence Processing Improve Sentence Readability Ranking." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 958–968. DOI: 10.18653/v1/E17-1090. URL: http://aclweb.org/anthology/E17-1090 (visited on 08/26/2018) (cit. on p. 92).

Howcroft, David M., Dietrich Klakow & Vera Demberg (Aug. 2017). "The Extended SPaRKy Restaurant Corpus: Designing a Corpus with Variable Information Density." In: *Proc. of Interspeech 2017*. Stockholm, Sweden: ISCA, pp. 3757–3761. DOI: 10.21437/Interspeech.2017-1555. URL: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1555.html (visited on 06/05/2018) (cit. on p. 80).

Howcroft, David M., Crystal Nakatsu & Michael White (2013). "Enhancing the Expression of Contrast in the SPaRKy Restaurant Corpus." In: *Proc. of the 14th European Workshop on Natural Language Generation (ENLG 2013)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 30–39. URL: https://aclanthology.org/W13-2104/ (cit. on pp. 78, 83, 94, 98).

Howcroft, David, Jorrig Vogels & Vera Demberg (2017). "G-TUNA: A Corpus of Referring Expressions in German, Including Duration Information." In: *Proceedings of the 10th International Conference on Natural Language Generation*. Santiago de Compostela, Spain: Association for Computational Linguistics, pp. 149–153. DOI: 10.18653/v1/W17-3522. URL: http://aclweb.org/anthology/W17-3522 (visited on 08/22/2018) (cit. on p. 161).

Hunter, James, Yvonne Freer, Albert Gatt, Ehud Reiter, Somayajulu Sripada & Cindy Sykes (Nov. 2012). "Automatic Generation of Natural Language Nursing Shift Summaries in Neonatal Intensive Care: BT-Nurse." In: *Artificial Intelligence in Medicine* 56.3, pp. 157–172. ISSN: 09333657. DOI: 10/f4j6j9. URL: https://linkinghub.elsevier.com/retrieve/pii/S0933365712001170 (visited on 05/27/2019) (cit. on pp. 94, 95).

Irurozki, Ekhine, Borja Calvo & Jose A. Lozano (2016). "PerMallows : An R Package for Mallows and Generalized Mallows Models." In: *Journal of Statistical Software* 71.12. ISSN: 1548-7660. DOI: 10/gf4775. URL: http://www.jstatsoft.org/v71/i12/ (visited on 07/19/2019) (cit. on p. 129).

Jones, Bevan Keeley, Mark Johnson & Sharon Goldwater (2012). "Semantic Parsing with Bayesian Tree Transducers." In: *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vol. Volume 1: Long Papers. Jeju, Republic of Korea: Association for Computational Linguistics, pp. 488–496 (cit. on p. 60).

Joshi, Aravind & Owen Rambow (2003). "A Formalism for Dependency Grammar Based on Tree Adjoining Grammar." In: *Proceedings of the Conference on Meaning-Text Theory*, pp. 207–216 (cit. on pp. vii, viii, 39, 42, 165).

Jurafsky, Daniel, Alan Bell, Michelle Gregory & William D Raymond (2001). "Probabilistic Relations between Words: Evidence from Reduction in Lexical Production." In: *Frequency and the emergence of linguistic structure*, pp. 229–254. ISSN: 9789027229472 (hbk., Eur.) DOI: 10.1075/tsl.45.13jur (cit. on p. 73).

Karpathy, Andrej & Li Fei-Fei (2015). "Deep Visual-Semantic Alignments for Generating Image Descriptions." In: *CVPR*, p. 10 (cit. on p. 90).

Khan, Imtiaz Hussain (June 2016). "Production of Referring Expressions in Arabic." In: *International Journal of Speech Technology* 19.2, pp. 385–392. ISSN: 1381-2416, 1572-8110. DOI: 10/gfw7k8. URL: http://link.springer.com/10.1007/s10772-015-9282-8 (visited on 03/19/2019) (cit. on p. 151).

Kiddon, Chloé, Luke Zettlemoyer & Yejin Choi (2016). "Globally Coherent Text Generation with Neural Checklist Models." In: *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Austin, Texas, USA: Association for Computational Linguistics, pp. 329–339 (cit. on pp. 20, 90).

Knight, Kevin & Vasileios Hatzivassiloglou (1995). "Two-Level, Many-Paths Generation." In: *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics* -. Cambridge, Massachusetts: Association for Computational Linguistics, pp. 252–260. DOI: 10/dzhvnt. URL: http://portal.acm.org/citation.cfm?doid=981658.981692 (visited on 04/09/2019) (cit. on pp. 17, 101).

Koller, Alexander & Jörg Hoffmann (2010). "Waking Up a Sleeping Rabbit: On Natural-Language Sentence Generation with FF." In: *Proc. of the 20th International Conference on Automated Planning and Scheduling (ICAPS)*. Ed. by Ronen I Brafman, Hector Geffner, Jörg Hoffmann & Henry A Kautz. AAAI, pp. 238–241 (cit. on pp. 16, 101).

Koller, Alexander & Marco Kuhlmann (2012). "Decomposing TAG Algorithms Using Simple Algebraizations." In: *Proc. of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+)*. Paris, France: Association for Computational Linguistics, pp. 135–143. URL: http://www.aclweb.org/anthology/W12-4616 (visited on 04/26/2017) (cit. on p. 67).

Koller, Alexander & Ronald P. A. Petrick (2011). "Experiences with Planning for Natural Language Generation." In: *Computational Intelligence* 27.1, pp. 23–40. ISSN: 08247935. DOI: 10.1111/j.1467-8640.2010.00370.x (cit. on p. 101).

Koller, Alexander & Matthew Stone (2007). "Sentence Generation as a Planning Problem." In: *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)* (cit. on p. 16).

Koolen, Ruud, Martijn Goudbeek & Emiel Krahmer (2011). "Effects of Scene Variation on Referential Overspecification." In: *CogSci*. URL: http://palm.mindmodeling.org/cogsci2011/papers/0234/paper0234.pdf (visited on 05/10/2017) (cit. on p. 160).

Koolen, Ruud & Emiel Krahmer (2010). "The D-TUNA Corpus: A Dutch Dataset for the Evaluation of Referring Expression Generation Algorithms." In: *LREC*. URL: http://tst-centrale.org/images/stories/producten/documentatie/dtuna_documentatie_en.pdf (visited on 05/10/2017) (cit. on pp. xi, 150, 168).

Kousidis, S, C Kennington, T Baumann, H Buschmeier, S Kopp & D Schlangen (Apr. 2014). "Situationally Aware In-Car Information Presentation Using Incremental Speech Generation: Safer, and More Effective." In: *Procs. of the EACL 2014 Workshop Dialogue In Motion*. Gothenburg, Sweden: ACL, pp. 68–72. URL: http://www.speech.kth.se/prod/publications/files/101718.download[1] (cit. on p. 97).

Krahmer, Emiel & Kees Van Deemter (2012). "Computational Generation of Referring Expressions: A Survey." In: *Computational Linguistics* 38.1, pp. 173–218. URL: http://www.mitpressjournals.org/doi/abs/10.1162/COLI_a_00088 (visited on 05/10/2017) (cit. on p. 150).

Krahmer, Emiel, Sebastiaan Van Erk & André Verleg (2003). "Graph-Based Generation of Referring Expressions." In: *Computational Linguistics* 29.1, pp. 53–72. URL: http://www.mitpressjournals.org/doi/abs/10.1162/089120103321337430 (visited on 05/11/2017) (cit. on p. 149).

Kuznetsova, Polina, Vicente Ordonez, Alexander Berg, Tamara Berg & Yejin Choi (July 2012). "Collective Generation of Natural Image Descriptions." In: *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*. Jeju, Republic of Korea: Association for Computational Linguistics, pp. 359–368 (cit. on pp. 93, 95).

Lavoie, Benoit & 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: Association for Computational Linguistics, pp. 265–268. DOI: 10.3115/974557.974596. URL: http://portal.acm.org/citation.cfm?doid=974557.974596 (visited on 09/10/2018) (cit. on p. 101).

Lebret, Remi, David Grangier & Michael Auli (Mar. 2016). "Neural Text Generation from Structured Data with Application to the Biography Domain." In: *Proc. of the Conference on Empirical Methods in NLP (EMNLP)*. Austin, Texas, USA: Association for Computational Linguistics. arXiv: 1603.07771 (cit. on p. 90).

Lester, James C & Bruce W Porter (1997). "Developing and Empirically Evaluating Robust Explanation Generators: The KNIGHT Experiments." In: *Computational Linguistics* 23.1, p. 38 (cit. on p. 94).

Lin, Chin-Yew & Eduard Hovy (2003). "Automatic Evaluation of Summaries Using N-Gram Co-Occurrence Statistics." In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*. Vol. 1. Edmonton, Canada: Association for Computational Linguistics, pp. 71–78. DOI: 10/dnpcb4. URL: http://portal.acm.org/citation.cfm?doid=1073445.1073465 (visited on 05/25/2019) (cit. on p. 91).

Lockwood, Patricia (Aug. 2021). *Official Diagnosis from Her Vet: Miette Most Likely Ate a Lizard and Tripped so Hard That She Lost Control of Her Body from the Neck down and Went Temporarily Blind for 36 Hours. Her Bloodwork Is Completely Normal but She Will Go Forth as Someone Who Has Seen God*. URL: https://twitter.com/tricialockwood/status/1428438388403810309 (cit. on p. 140).

Lukin, Stephanie, Lena Reed & Marilyn Walker (2015). "Generating Sentence Planning Variations for Story Telling." In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Prague, Czech Republic: Association for Computational Linguistics, pp. 188–197. DOI: 10.18653/v1/W15-4627. URL: http://aclweb.org/anthology/W15-4627 (visited on 08/15/2018) (cit. on p. 17).

Mahapatra, Joy, Sudip Kumar Naskar & Sivaji Bandyopadhyay (2016). "Statistical Natural Language Generation from Tabular Non-Textual Data." In: *The 9th International Conference on Natural Language Generation*. Edinburgh, Scotland, UK: Association for Computational Linguistics, pp. 143–152 (cit. on p. 19).

Mahr, Angela, Michael Feld, Mohammad Mehdi Moniri & Rafael Math (2012). "The ConTRe (Continuous Tracking and Reaction) Task: A Flexible Approach for Assessing Driver Cognitive Workload with High Sensitivity Categories and Subject Descriptors." In: *International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI)*. Ed. by Andrew L. Kun, Linda Ng Boyle, Bryan Reimer & Andreas Riener. Portsmouth, NH, USA, pp. 88–91. URL: http://www.dfki.de/web/forschung/iwi/publikationen/renameFileForDownload?filename=ConTRe_WS.pdf&file_id=uploads_1846 (cit. on p. 151).

Mairesse, Fran¸cois, Milica Gǎsić, Filip Jurčĭcek, Simon Keizer, Blaise Thomson, Kai Yu & Steve Young (2010). "Phrase-Based Statistical Language Generation Using Graphical Models and Active Learning." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Uppsala, Sweden: Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/P10-1157 (cit. on pp. 18, 74).

Mairesse, Fran¸cois & Marilyn a. Walker (2011). "Controlling User Perceptions of Linguistic Style: Trainable Generation of Personality Traits." In: *Computational Linguistics* 37.January 2009, pp. 455–488. ISSN: 0891-2017. DOI: 10.1162/COLI_a₀0063 (cit. on p. 94).

Mairesse, Fran¸cois & Steve Young (2014). "Stochastic Language Generation in Dialogue Using Factored Language Models." In: *Computational Linguistics* 40.4. ISSN: 04194217. DOI: 10.1162/COLI_a₀0199 (cit. on p. 18).

Mairesse, François & Marilyn A. Walker (2007). "PERSONAGE : Personality Generation for Dialogue." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 496–503. URL: https://aclanthology.org/P07-1063 (cit. on p. 94).

Mani, Inderjeet, David House, Gary Klein, Lynette Hirschman, Therese Firmin & Beth Sundheim (1999). "The TIPSTER SUMMAC Text Summarization Evaluation." In: *Proc. of EACL '99*. Association for Computational Linguistics, pp. 77–85. DOI: 10.3115/977035.977047. URL: http://portal.acm.org/citation.cfm?doid=977035.977047 (visited on 04/11/2018) (cit. on p. 96).

Mann, William C & Sandra A Thompson (1988). "Rhetorical Structure Theory: Towards a Functional Theory of Text Organization." In: *TEXT* 8.3, pp. 243–281 (cit. on pp. 77–79).

Marcu, Daniel (1997). "From Local to Global Coherence: A Bottom-up Approach to Text Planning." In: *Proc. of AAAI-97*. AAAI (cit. on p. 16).

Marshall, Sandra P. (2002). "The Index of Cognitive Activity: Measuring Cognitive Workload." In: *Proceedings of the IEEE 7th Conference*

on Human Factors and Power Plants, pp. 5–9. ISSN: 07350015. DOI: 10.1109/HFPP.2002.1042860 (cit. on p. 151).

Mellish, Chris, Alistair Knott, Jon Oberlander & Mick O'Donnell (1998). "Experiments Using Stochastic Search for Text Planning." In: *Proc. of the 9th International Workshop on Natural Language Generation (INLG)*. Niagra-on-the-Lake, Ontario, Canada: Association for Computational Linguistics (cit. on p. 16).

Meteer, Marie W (June 1990). "Abstract Linguistic Resources for Text Planning." In: *Proc. of the 5th International Workshop on Natural Language Generation (INLG)*. Dawson, Pennsylvania, USA: Association for Computational Linguistics, pp. 62–69 (cit. on p. 15).

Mille, Simon, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler & Leo Wanner (2018). "The First Multilingual Surface Realisation Shared Task (SR'18): Overview and Evaluation Results." In: *Proceedings of the First Workshop on Multilingual Surface Realisation*. Melbourne, Australia: Association for Computational Linguistics, pp. 1–12. DOI: 10/gmh7n4. URL: http://aclweb.org/anthology/W18-3601 (visited on 08/05/2021) (cit. on pp. 64, 169).

Mille, Simon, Anja Belz, Bernd Bohnet, Yvette Graham & Leo Wanner (2019). "The Second Multilingual Surface Realisation Shared Task (SR'19): Overview and Evaluation Results." In: *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 1–17. DOI: 10/gg5wff. URL: https://www.aclweb.org/anthology/D19-6301 (visited on 08/05/2021) (cit. on pp. 64, 169).

Mille, Simon, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham & Leo Wanner (Dec. 2020). "The Third Multilingual Surface Realisation Shared Task (SR'20): Overview and Evaluation Results." In: *Proc. of the 3rd Workshop on Multilingual Surface Realisation (MSR)*. Barcelona, Spain (Online): Association for Computational Linguistics, pp. 1–20. URL: https://aclanthology.org/2020.msr-1.1/ (cit. on pp. 64, 169).

Mitchell, Margaret, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alex Berg, Tamara Berg & Hal Daume Iii (Aug. 2012). "Midge: Generating Image Descriptions From Computer Vision Detections." In: *Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Avignon, France: Association for Computational Linguistics, pp. 747–756 (cit. on pp. 92–95).

Nakatsu, Crystal & Michael White (2010). "Generating with Discourse Combinatory Categorial Grammar." In: *Language Issues in Language Technology* 4.September, pp. 1–62. URL: http://elanguage.net/journals/index.php/lilt/article/viewArticle/1277 (cit. on pp. 13, 66).

Nayak, Neha, Dilek Hakkani-Tur, Marilyn Walker & Larry Heck (2017). "To Plan or Not to Plan? Discourse Planning in Slot-Value In-

formed Sequence to Sequence Models for Language Generation." In: *Proc. of Interspeech* (cit. on p. 20).

Novikova, Jekaterina, Ondřej Dušek & Verena Rieser (2017). "The E2E Dataset: New Challenges For End-to-End Generation." In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Saarbrücken, Germany: Association for Computational Linguistics, pp. 201–206. DOI: 10.18653/v1/W17-5525. URL: http://aclweb.org/anthology/W17-5525 (visited on 08/16/2018) (cit. on pp. 76, 81, 84, 93–95).

Novikova, Jekaterina, Ondřej Dušek & Verena Rieser (2018). "RankME: Reliable Human Ratings for Natural Language Generation." In: *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (NAACL-HLT)*. New Orleans, Louisiana, USA: Association for Computational Linguistics, pp. 72–78. DOI: 10.18653/v1/N18-2012 (cit. on pp. 94, 95, 98, 99).

Novikova, Jekaterina, Oliver Lemon & Verena Rieser (2016). "Crowd-Sourcing NLG Data: Pictures Elicit Better Data." In: *The 9th International Conference on Natural Language Generation*. Edinburgh, Scotland, UK: Association for Computational Linguistics, pp. 265–273. URL: https://aclweb.org/anthology/W16-6644/ (cit. on pp. 76, 81, 83).

Oberlander, Jon & Chris Brew (2000). "Stochastic Text Generation." In: *Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 358.1769, pp. 1373–1387 (cit. on pp. 17, 19).

Oh, Alice H. & Alexander I. Rudnicky (2002). "Stochastic Natural Language Generation for Spoken Dialog Systems." In: *Computer Speech & Language* 16.3-4, pp. 387–407. ISSN: 08852308. DOI: 10.1016/S0885-2308(02)00012-8. URL: http://www.sciencedirect.com/science/article/pii/S0885230802000128 (cit. on pp. 17, 19, 94, 95).

Oraby, Shereen, Lena Reed, Shubhangi Tandon, T. S. Sharath, Stephanie Lukin & Marilyn Walker (July 2018). "Controlling Personality-Based Stylistic Variation with Neural Natural Language Generators." In: *Proc. of the SIGDIAL 2018 Conference*. Melbourne, Australia: Association for Computational Linguistics, pp. 180–190. DOI: 10.18653/v1/W18-5019. arXiv: 1805.08352. URL: https://aclanthology.org/W18-5019/ (visited on 05/28/2018) (cit. on p. 94).

Palmer, Martha, Daniel Gildea & Paul Kingsbury (Mar. 2005). "The Proposition Bank: An Annotated Corpus of Semantic Roles." In: *Computational Linguistics* 31.1, pp. 71–106. ISSN: 0891-2017. DOI: 10.1162/0891201053630264. URL: http://www.mitpressjournals.org/doi/abs/10.1162/0891201053630264 (cit. on p. 16).

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu (2002). "BLEU: A Method for Automatic Evaluation of Machine Translation." In: *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: http://portal.acm.org/citation.cfm?doid=1073083.1073135 (visited on 04/24/2018) (cit. on pp. xii, 89).

Pietsch, Johannes (Feb. 2017). "Learning Lexicalization Rules for Surface Realization in an NLG System." PhD thesis. Saarbruecken, Germany: Saarland University (cit. on p. 118).

Pitman, Jim & Marc Yor (1997). "The Two-Parameter Poisson-Dirichlet Distribution Derived from a Stable Subordinator." In: *The Annals of Probability* 25.2, pp. 855–900. DOI: 10/dc4tdx (cit. on p. 52).

Portet, François, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer & Cindy Sykes (May 2009). "Automatic Generation of Textual Summaries from Neonatal Intensive Care Data." In: *Artificial Intelligence* 173.7-8, pp. 789–816. ISSN: 00043702. DOI: 10/cfktxh. URL: https://linkinghub.elsevier.com/retrieve/pii/S0004370208002117 (visited on 06/16/2019) (cit. on p. 96).

Post, Matt & Daniel Gildea (2009a). "Bayesian Learning of a Tree Substitution Grammar." In: *Proc. of the ACL-IJCNLP 2009 Conference*. Vol. Short Papers. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/P/P09/P09-2012%5Cnhttp://dl.acm.org/citation.cfm?id=1667599 (cit. on p. 55).

Post, Matt & Daniel Gildea (2009b). "Language Modeling with Tree Substitution Grammars." In: *NIPS Workshop on Grammar Induction, Representation of Language, and Language Learning*. URL: http://www.cs.rochester.edu/gildea/pubs/post-gildea-nips09.pdf (cit. on p. 55).

Pusse, Florian, Asad Sayeed & Vera Demberg (June 2016). "LingoTurk: Managing Crowdsourced Tasks for Psycholinguistics." In: *Proc. of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (Demonstrations)*. San Diego, California, USA: Association for Computational Linguistics, pp. 57–61 (cit. on pp. 81, 104).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei & Ilya Sutskever (2019). *Language Models Are Unsupervised Multitask Learners*. Tech. rep. (cit. on p. 63).

Rajkumar, Rajakrishnan, Dominic Espinosa & Michael White (2011). "The OSU System for Surface Realization at Generation Challenges 2011." In: *Proc. of the 13th European Workshop on Natural Language Generation (ENLG)*. Nancy, France: Association for Computational Linguistics, pp. 236–238 (cit. on p. 90).

Rajkumar, Rajakrishnan, Michael White & Dominic Espinosa (June 2009). "Exploiting Named Entity Classes in CCG Surface Realization." In: *LREC*, p. 161. DOI: 10.3115/1620853.1620898. URL: http://portal.acm.org/citation.cfm?doid=1620853.1620898 (cit. on p. 90).

Rambow, Owen, Srinivas Bangalore & Marilyn A. Walker (2001). "Natural Language Generation in Dialog Systems." In: *Proc. of the 1st International Conference on Human Language Technology Research (HLT)*. San Diego, California, USA: Association for Computational Linguistics (cit. on pp. 17, 19).

Ratnaparkhi, Adwait (July 2002). "Trainable Approaches to Surface Natural Language Generation and Their Application to Conversational Dialog Systems." In: *Computer Speech & Language* 16.3-4, pp. 435–455. ISSN: 08852308. DOI: 10/cz4cqb. URL: http://linkinghub.elsevier.com/retrieve/pii/S0885230802000256 (visited on 01/22/2018) (cit. on p. 19).

Raymond, William D., Robin Dautricourt & Elizabeth Hume (2006). "Word-Internal /t,d/ Deletion in Spontaneous Speech: Modeling the Effects of Extra-Linguistic, Lexical, and Phonological Factors." In: *Language Variation and Change* 18, pp. 55–97. ISSN: 0954-3945. DOI: 10.1017/S0954394506060042 (cit. on p. 73).

Reed, Lena, Shereen Oraby & Marilyn Walker (2018). "Can Neural Generators for Dialogue Learn Sentence Planning and Discourse Structuring?" In: *Proc. of the 11th International Conference on Natural Language Generation (INLG)*. Association for Computational Linguistics (cit. on p. 20).

Reiter, Ehud (June 2018). "A Structured Review of the Validity of BLEU." In: *Computational Linguistics*, pp. 1–8. ISSN: 0891-2017, 1530-9312. DOI: 10/gdnfq7. URL: https://www.mitpressjournals.org/doi/abs/10.1162/COLI_a_00322 (visited on 06/20/2018) (cit. on p. 90).

Reiter, Ehud & Robert Dale (2000). *Building Natural Language Generation Systems*. Cambridge University Press. ISBN: 978-0-511-51985-7 (cit. on p. 11).

Reiter, Ehud, Roma Robertson & Liesl M. Osman (Mar. 2003). "Lessons from a Failure: Generating Tailored Smoking Cessation Letters." In: *Artificial Intelligence* 144.1-2, pp. 41–58. ISSN: 00043702. DOI: 10/dtsr6n. URL: https://linkinghub.elsevier.com/retrieve/pii/S0004370202003703 (visited on 05/28/2019) (cit. on p. 96).

Scha, Remko (1990). "Taaltheorie en taaltechnologie; competence en performance." In: *Computertoepassingen in de Neerlandistiek*. Ed. by R. de Kort & G. L. J. Leerdam. Almere, the Netherlands: Landelijke Vereniging Voor Neerlandici, pp. 7–22. URL: http://www.remkoscha.nl/Leerdam.html (visited on 05/12/2019) (cit. on p. 56).

Scholman, Merel & Vera Demberg (2017). "Crowdsourcing Discourse Interpretations: On the Influence of Context and the Reliability of a Connective Insertion Task." In: *Proceedings of the 11th Linguistic Annotation Workshop*. Valencia, Spain: Association for Computational Linguistics, pp. 24–33. DOI: 10/gf85q7. URL: http://aclweb.org/anthology/W17-0803 (visited on 10/01/2019) (cit. on p. 139).

Scholman Merel C. J., Scholman (Oct. 2018). "Coherence Relations in Discourse and Cognition: Comparing Approaches, Annotations, and Interpretations." PhD thesis. Saarbrücken, Germany: Saarland University (cit. on p. 171).

Schwenger, Maximilian, Alvaro Torralba, Joerg Hoffmann, David M Howcroft & Vera Demberg (Dec. 2016). "From OpenCCG to AI Planning: Detecting Infeasible Edges in Sentence Generation." In: *Proc. of the 26th International Conference on Computational Linguistics (COLING): Technical Papers*. Osaka, Japan: Association for Computational Linguistics, pp. 1524–1534 (cit. on p. 101).

Sha, Lei, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang & Zhifang Sui (2018). "Order-Planning Neural Text Generation From Structured Data." In: p. 8 (cit. on p. 90).

Shannon, Claude E. & W. Weaver (1948). "A Mathematical Theory of Communication." In: *The Bell System Technical Journal* 27.3, pp. 379–423. ISSN: 1559-1662. DOI: 10.1145/584091.584093. URL: http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf (cit. on pp. 3, 73).

Shen, Sheng, Daniel Fried, Jacob Andreas & Dan Klein (Apr. 2019). "Pragmatically Informative Text Generation." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Vol. Long & Short Papers. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 4060–4067. arXiv: 1904.01301. URL: http://arxiv.org/abs/1904.01301 (visited on 04/16/2019) (cit. on p. 90).

Shindo, Hiroyuki, Akinori Fujino & Masaaki Nagata (2011). "Insertion Operator for Bayesian Tree Substitution Grammars." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*. Association for Computational Linguistics, pp. 206–211 (cit. on p. 55).

Shindo, Hiroyuki, Yusuke Miyao, Akinori Fujino & Masaaki Nagata (2012). "Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing." In: *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL): Long Papers*. Vol. 1. Association for Computational Linguistics, pp. 440–448. URL: http://dl.acm.org/citation.cfm?id=2390586 (visited on 08/11/2017) (cit. on p. 55).

Snover, Matthew, Bonnie Dorr, Richard Schwartz, John Makhoul, Linnea Micciulla & Ralph Weischedel (July 2005). *A Study of Translation Error Rate with Targeted Human Annotation*. Tech. rep. (cit. on p. 91).

Sripada, Somayajulu G, Ehud Reiter & Lezan Hawizy (2005). "Evaluation of an NLG System Using Post-Edit Data: Lessons Learnt." In: *Proc. of the 10th European Workshop on Natural Language Generation (ENLG)*, p. 7 (cit. on p. 94).

Steedman, M. & J. Baldridge (2006). "Combinatory Categorial Grammar." In: *Encyclopedia of Language & Linguistics (Second Edition)*. Ed. by Keith Brown. Second Edition. Oxford: Elsevier, pp. 610–621. ISBN: 978-0-08-044854-1. DOI: 10.1016/B0-08-044854-2/02028-9. URL: https://www.sciencedirect.com/science/article/pii/B0080448542020289 (cit. on p. 65).

Stent, Amanda & Martin Molina (2009). "Evaluating Automatic Extraction of Rules for Sentence Plan Construction." In: *Proc. of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. London, England, UK: Association for Computational Linguistics, pp. 290–297. URL: http://www.aclweb.org/anthology/W09-3941 (cit. on pp. 16, 17).

Stent, Amanda, Rashmi Prasad & Marilyn Walker (2004). "Trainable Sentence Planning for Complex Information Presentation in Spoken Dialog Systems." In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*. Barcelona, Spain: Association for Computational Linguistics, 79–es. DOI: 10.3115/1218955.1218966. URL: http://portal.acm.org/citation.cfm?doid=1218955.1218966 (visited on 08/15/2018) (cit. on pp. 16, 92, 93, 95).

Stone, Matthew & Christine Doran (1997). "Sentence Planning as Description Using Tree Adjoining Grammar." In: *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)* (cit. on p. 15).

Stone, Matthew, Christine Doran, Bonnie Webber, Tonia Bleam & Martha Palmer (2003). "Microplanning with Communicative Intentions: The SPUD System." In: *Computational Intelligence* 19.4, pp. 311–381. URL: https://www.cs.rutgers.edu/~mdstone/pubs/spudr.pdf (visited on 09/21/2018) (cit. on p. 15).

Takase, Sho, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao & Masaaki Nagata (2016). "Neural Headline Generation on Abstract Meaning Representation." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1054–1059. DOI: 10/gfztdc. URL: http://aclweb.org/anthology/D16-1112 (visited on 04/24/2019) (cit. on p. 90).

Tran, Van-Khanh & Le-Minh Nguyen (2017). "Natural Language Generation for Spoken Dialogue System Using RNN Encoder-Decoder Networks." In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 442–451. DOI: 10/gfzhw9. URL: http://aclweb.org/anthology/K17-1044 (visited on 04/16/2019) (cit. on p. 90).

Tran, Van-Khanh, Le-Minh Nguyen & Satoshi Tojo (2017). "Neural-Based Natural Language Generation in Dialogue Using RNN Encoder-Decoder with Semantic Aggregation." In: Association for Computational Linguistics, pp. 231–240. DOI: 10.18653/v1/W17-5528. URL: http://aclweb.org/anthology/W17-5528 (visited on 04/24/2018) (cit. on p. 90).

Vogels, Jorrig, David M. Howcroft, Elli Tourtouri & Vera Demberg (2020). "How Speakers Adapt Object Descriptions to Listeners under Load." In: *Language, Cognition and Neuroscience* 35.1, pp. 78–92. DOI: 10.1080/23273798.2019.1648839. URL: https://www.tandfonline.com/doi/full/10.1080/23273798.2019.1648839 (cit. on p. 158).

Walker, Marilyn A., Owen Rambow & Monica Rogati (2001). "SPoT: A Trainable Sentence Planner." In: *Proc. of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*. Pittsburgh, Pennsylvania, USA: Association for Computational Linguistics. DOI: 10.3115/1073336.1073339. URL: http://portal.acm.org/citation.cfm?doid=1073336.1073339 (visited on 09/07/2018) (cit. on p. 16).

Walker, Marilyn A., Amanda Stent, François Mairesse & Rashmi Prasad (2007). "Individual and Domain Adaptation in Sentence Planning for Dialogue." In: *Journal of Artificial Intelligence Research* 30, pp. 413–456. ISSN: 10769757. DOI: 10.1613/jair.2329. URL: https://jair.org/index.php/jair/article/view/10519/25195 (cit. on pp. ix, 16, 77, 134).

Walker, Marilyn A., Steve J. Whittaker, Amanda Stent, P. Maloor, Johanna D. Moore, M. Johnston & G. Vasireddy (Oct. 2004). "Generation and Evaluation of User Tailored Responses in Multimodal Dialogue." In: *Cognitive Science* 28.5, pp. 811–840. ISSN: 03640213. DOI: 10.1016/j.cogsci.2004.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0364021304000667 (cit. on p. 77).

Wen, Tsung-Hsien "Shawn", Milica Gǎsić, Nikola Mrǩsić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke & Steve Young (2016). "Multi-Domain Neural Network Language Generation for Spoken Dialogue Systems." In: *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. San Diego, California, USA: Association for Computational Linguistics. DOI: 10.

18653/v1/N16-1015. URL: https://aclweb.org/anthology/N16-1015/ (cit. on p. 74).

Wen, Tsung-Hsien "Shawn", Pei-hao Su, David Vandyke, Steve Young & Trumpington Street (Sept. 2015a). "Semantically Conditioned LSTM-Based Natural Language Generation for Spoken Dialogue Systems." In: *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1711–1721. DOI: 10.18653/v1/D15-1199. URL: http://www.aclweb.org/anthology/D15-1199 (cit. on pp. 19, 20).

Wen, Tsung-Hsien, Milica Găsić, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke & Steve Young (Sept. 2015b). "Stochastic Language Generation in Dialogue Using Recurrent Neural Networks with Convolutional Sentence Reranking." In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Prague, Czech Republic: Association for Computational Linguistics, pp. 275–284. DOI: 10/gfzhw5. URL: http://aclweb.org/anthology/W15-4639 (visited on 04/16/2019) (cit. on pp. 19, 20, 74, 76).

White, Michael (2004). "Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar." In: *Research on Language and Computation* 2006.4, pp. 39–75. URL: http://www.ling.ohio-state.edu/Âămwhite/papers/White-RoLC-2004-to-appear.pdf (cit. on pp. 65, 101).

White, Michael & David M. Howcroft (2015). "Inducing Clause-Combining Rules: A Case Study with the SPaRKy Restaurant Corpus." In: *Proc. of the 15th European Workshop on Natural Language Generation (ENLG)*. Brighton, United Kingdom: Association for Computational Linguistics, pp. 28–37. DOI: 10.18653/v1/W15-4704. URL: http://aclweb.org/anthology/W15-4704 (cit. on p. 102).

Wiseman, Sam, Stuart M. Shieber & Alexander M. Rush (Aug. 2018). "Learning Neural Templates for Text Generation." arXiv: 1808.10122. URL: http://arxiv.org/abs/1808.10122 (visited on 04/24/2019) (cit. on p. 90).

Xiao, Tong & Jingbo Zhu (2013). "Unsupervised Sub-Tree Alignment for Tree-to-Tree Translation." In: *Journal of Artificial Intelligence Research* 48, pp. 733–782. URL: http://www.jair.org/papers/paper4033.html (visited on 08/09/2017) (cit. on p. 60).

Xu, Wei, Courtney Napoles, Ellie Pavlick, Quanze Chen & Chris Callison-Burch (Dec. 2016). "Optimizing Statistical Machine Translation for Text Simplification." In: *Transactions of the Association for Computational Linguistics* 4, pp. 401–415. ISSN: 2307-387X. DOI: 10/gf24vz. URL: https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00107 (visited on 05/25/2019) (cit. on p. 91).

Yamangil, Elif & Stuart M. Shieber (2010). "Bayesian Synchronous Tree-Substitution Grammar Induction and Its Application to Sen-

tence Compression." In: *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, pp. 937–947. URL: http://dl.acm.org/citation.cfm?id=1858777 (visited on 08/09/2017) (cit. on pp. 51, 56–59).

Yamangil, Elif & Stuart M. Shieber (2013). "Nonparametric Bayesian Inference and Efficient Parsing for Tree-Adjoining Grammars." In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (cit. on p. 55).

Young, R.Michael (Dec. 1999). "Using Grice's Maxim of Quantity to Select the Content of Plan Descriptions." In: *Artificial Intelligence* 115.2, pp. 215–256. ISSN: 00043702. DOI: 10/b532cs. URL: https://linkinghub.elsevier.com/retrieve/pii/S000437029900082X (visited on 05/28/2019) (cit. on p. 96).

Young, Steve (2009). *CUED Standard Dialogue Acts*. Tech. rep., pp. 1–12. URL: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:CUED+Standard+Dialogue+Acts#0http://mi.eng.cam.ac.uk/Âăsjy/papers/youn09.pdf (cit. on pp. 17, 72).

Zarrieß, Sina, Julian Hough, Casey Kennington, Ramesh Manuvinakurike, David DeVault, Raquel Fernandez & David Schlangen (2016). "PentoRef: A Corpus of Spoken References in Task-Oriented Dialogues." In: *10th Edition of the Language Resources and Evaluation Conference*. URL: https://pub.uni-bielefeld.de/download/2903076/2903079 (visited on 07/17/2017) (cit. on p. 151).

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre & Daniel Zeman (May 2021). "Universal Dependencies." In: *Computational Linguistics*, pp. 1–54. ISSN: 0891-2017, 1530-9312. DOI: 10/gjmqqp. URL: https://direct.mit.edu/coli/article/doi/10.1162/coli_a_00402/98516/Universal-Dependencies (visited on 08/10/2021) (cit. on pp. 33, 169).

tarrlab (1996). *The Object Databank*. URL: https://sites.google.com/andrew.cmu.edu/tarrlab/resources/tarrlab-stimuli (cit. on pp. 150, 156).

van Deemter, Kees (2002). "Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm." In: *Computational Linguistics*. DOI: 10.1162/089120102317341765. URL: http://homepages.abdn.ac.uk/k.vdeemter/refex-cl.pdf (visited on 05/11/2017) (cit. on p. 149).

van Deemter, Kees, Le Sun, Rint Sybesma, Xiao Li, Chen Bo & Muyun Yang (2017). "Investigating the Content and Form of Referring Expressions in Mandarin: Introducing the Mtuna Corpus." In: *Proceedings of the 10th International Conference on Natural Language Generation*. Santiago de Compostela, Spain: Association for Computational Linguistics, pp. 213–217. DOI: 10/gfw7k9. URL: http:

//aclweb.org/anthology/W17-3532 (visited on 03/19/2019) (cit. on p. 151).

van Deemter, Kees, Ielka van der Sluis & Albert Gatt (2006). "Building a Semantically Transparent Corpus for the Generation of Referring Expressions." In: *Proceedings of the Fourth International Natural Language Generation Conference*. Association for Computational Linguistics, pp. 130–132. URL: http://dl.acm.org/citation.cfm?id=1706296 (visited on 05/09/2017) (cit. on pp. xi, 150, 168).

DECLARATION

I hereby declare that I composed this thesis entirely myself and that it describes my own research. I have not used any literature or materials other than the ones referred to in this thesis.

ERKLÄRUNG

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig verfasst habe, und dass es meine eigene Forschung beschreibt. Keine anderen als die angegebenen Quellen und Hilfsmittel sind verwendet. Declaration

Edinburgh, Scotland, UK, August 2021

David M. Howcroft

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/