DSNNLG 2019

1st Workshop on Discourse Structure in Neural NLG

Proceedings of the Workshop

November 1, 2019
Tokyo, Japan

Endorsed by SIGGEN and SIGDIAL

Sponsored by Saarland University (SFB1102 and SFB-TR 248) and The Ohio State University

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-950737-67-3

Introduction

Welcome to the 1st Workshop on Discourse Structure in Neural NLG, a workshop held in conjunction with INLG 2019, the International Conference on Natural Language Generation, in Tokyo, Japan.

Neural methods for natural language generation (NNLG) arrived with much fanfare a few years ago and became the dominant method employed in the E2E NLG Challenge. While neural methods promise flexible, end-to-end trainable models, recent studies have revealed their inability to produce satisfactory output for longer or more complex texts as well as how the black-box nature of these models makes them difficult to control, in contrast to traditional NLG architectures that make use of explicit representations of discourse structure and/or sentence planning operations. As such, several papers have recently appeared that explore how to incorporate intermediate structures into NNLG or otherwise improve coherence and cohesion.

This workshop aims to encourage further research on enhancing quality in NNLG in terms of discourse coherence and cohesion along with ways to make NNLG models easier to control. Topics covered will include the limits of current end-to-end NNLG with respect to sentence planning and discourse structure; methods for improving discourse coherence and cohesion in NNLG, for example by making better use of discourse connectives, or by avoiding unnecessary repetition; methods for control and interpretability of NNLG, for example by providing more explicit guidance or structure in the input; and better methods for evaluating discourse coherence and cohesion in NNLG.

These proceedings include a total of four papers, chosen from seven submitted papers, each reviewed by three members of the program committee. In addition to presentation of papers, the workshop will host four invited talks by Thiago Castro Ferreira, Angela Fan, Behnam Hedayatnia and Lu Wang, as well as three non-archival presentations, and a panel on remaining challenges.

We would like to thank everyone who contributed to the success of this workshop, especially the authors, the program committee members, the organizers of the INLG 2019 conference and the INLG 2019 workshop chairs, and our sponsors, Saarland University and The Ohio State University.

—The Organizers

Organizers:

Anusha Balakrishnan, Microsoft Semantic Machines
Vera Demberg, Saarland University
Chandra Khatri, Uber AI
Abhinav Rastogi, Google AI
Donia Scott, University of Sussex / Scott Rush Associates
Marilyn Walker, University of California, Santa Cruz
Michael White, The Ohio State University / Facebook AI

Program Committee:

Paul Crook, Facebook AI
Alessandra Cervone, University of Trento
Claire Gardent, French National Centre for Scientific Research (CNRS)
Behnam Hedayatnia, Amazon Alexa AI
Dave Howcroft, Heriot-Watt University
Emiel Krahmer, Tillburg University
Shereen Oraby, University of California, Santa Cruz
Cecile Paris, CSIRO
Owen Rambow, Elemental Cognition
Alexander Rush, Harvard University
Frank Schilder, Thomson Reuters
Rajen Subba, Facebook AI
David Winer, University of Utah
Sam Wiseman, Toyota Technological Institute at Chicago
Amir Zeldes, Georgetown University
Yi-Chia Wang, Uber AI

Invited Speakers:

Thiago Castro Ferreira, Tilburg center for Cognition and Communication (TiCC), Tilburg University / Department of Linguistics, Federal University of Minas Gerais
Angela Fan, Facebook AI
Behnam Hedayatnia, Amazon Alexa AI
Lu Wang, CCIS, Northeastern University

Table of Contents

Invited Talks

DSNNLG Papers

Conference Program

10:00–11:30 Session 1

10:00–10:15 Welcome and Opening Remarks

10:15–11:00 *Data-to-text Natural Language Generation: Traditional, Novel and Future Methods* (Invited Talk)
Thiago Castro Ferreira

11:00–11:30 *Maximizing Stylistic Control and Semantic Accuracy in NLG: Personality Variation and Discourse Contrast*
Vrindavan Harrison, Lena Reed, Shereen Oraby and Marilyn Walker

11:30–12:00 Coffee Break

12:00–12:45 Session 2

12:00–12:45 *Convince Me If You Can: Argument Generation with Content Planning and Style Specification* (Invited Talk)
Lu Wang

12:45–2:15 Lunch

2:15–3:20 Session 3

2:15–3:00 *Hierarchical Structure in Story Generation* (Invited Talk)
Angela Fan

3:00–3:20 *Incorporating Textual Evidence in Visual Storytelling*
Tianyi Li and Sujian Li

3:20–4:15 Coffee Break and Poster Session

3:20–4:15 DSNNLG and Non-archival Posters
(continued next page)

4:15–5:45 Session 5

4:15–5:00 *Topical Chat: On the Structure of Knowledge Grounded Conversations* (Invited Talk)
Behnam Hedayatnia

5:00–5:45 Panel

Conference Program (cont.)

3:20–4:15 Coffee Break and Poster Session

DSNNLG Posters

Fine-Grained Control of Sentence Segmentation and Entity Positioning in Neural NLG
Kritika Mehta, Raheel Qader, Cyril Labbe and François Portet

Zero-shot Chinese Discourse Dependency Parsing via Cross-lingual Mapping
Yi Cheng and Sujian Li

Non-archival Posters

Towards a Scalable & Controllable Computational Solution for Document Planning
Craig Thomson, Ehud Reiter and Somayajulu Sripada

Modeling Conversation Context by Adapting Cognitive Architectures
Sashank Santhanam and Samira Shaikh

Constrained Decoding and Query Attention for Neural NLG in Task-Oriented Dialogue
Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White and Rajen Subba

Invited Talk

Data-to-text Natural Language Generation: Traditional, Novel and Future Methods

Thiago Castro Ferreira
Tilburg center for Cognition and Communication (TiCC), Tilburg University
Department of Linguistics, Federal University of Minas Gerais

Data-to-text Natural Language Generation (NLG) is a consolidated field of research which normally combines Computational Linguistics, Software Engineering and Artificial Intelligence methods to generate natural language from non-linguistic representations. Traditionally, most data-to-text applications have been designed using a modular pipeline architecture, in which the non-linguistic input data is converted into natural language through several intermediate transformations. In contrast, influenced by the phenomenon of deep learning, recent neural models for data-to-text generation have been proposed as end-to-end approaches, where the non-linguistic input is rendered in natural language with much less explicit intermediate representations in-between. Theoretically, we know that pipeline approaches are more transparent and their modules can be reused across applications, whereas neural end-to-end approaches may demand less manual labor and has registered state-of-the-art results in other text generation tasks, such as Machine Translation. Although we know these pros and cons in theory, the question about which kind of model empirically generates the most fluent and semantic texts from non-linguistic representations still remains unanswered. This lack of an empirical comparison is partially caused by the fact that traditional benchmarks for the task only consist of raw non-linguistic representations in parallel with their textual realizations, benefiting the evaluation of end-to-end approaches but not of pipeline ones, since explicit intermediate representations are missing for the study of particular modules of the latter architecture. In this presentation, I will introduce an annotation framework to enrich popular benchmarks with explicit intermediate representations, which will help the development and evaluation of particular pipeline modules as Discourse Ordering, Aggregation, Lexicalization, Referring Expressions Generation and Textual Realization. Based on a version of a popular data-to-text benchmark enriched with our framework, I will also present the results of a comparison between pipeline and end-to-end approaches. Finally, based on the findings of this project, I will project future challenges in the research of data-to-text NLG in terms of data, applications and evaluation.

Invited Talk

Convince Me If You Can: Argument Generation with Content Planning and Style Specification

Lu Wang
CCIS, Northeastern University

Understanding, evaluating, and generating arguments are crucial elements of the decision-making and reasoning process. A multitude of arguments and counter-arguments are constructed on a daily basis to persuade and inform us on a wide range of issues. However, constructing persuasive arguments is a challenging task for both human and computers, as it requires credible evidence, rigorous logical reasoning, and sometimes emotional appeals.

In this talk, I will introduce our neural network-based argument generation model. It consists of a powerful retrieval system and a novel two-step generation model, where a text planning decoder first decides on the main talking points and a proper language style for each sentence, then a content realization component constructs an informative and fluent paragraph-level argument. We believe that the proposed argument generation framework will enable many compelling applications, including providing unbiased perspectives on complex issues, debate coaching, and essay writing tutoring. Our framework is also generic and has been applied to other text generation problems, such as Wikipedia article paragraph generation and scientific paper abstract writing.

Invited Talk

Hierarchical Structure in Story Generation

Angela Fan
Facebook AI

We explore the task of story generation: creative systems that can build coherent and fluent passages of text about a topic. Using a dataset of 300k human-written stories paired with writing prompts, we investigate hierarchical story generation. Our models first generate a premise and then transform it into a passage of text. We develop models that improve the relevance of the story to the premise using a novel form of model fusion and present improvements to self-attention that better capture long-range context. Then, we build upon this work by proposing a coarse to fine mechanism for story generation, decomposing the task into several steps. We first explicitly generate logical verb sequences to model action in stories, then form these into sentences, and finally fill-in character names. We show that such decompositions improve the consistency and diversity of generated stories.

Invited Talk

Topical Chat: On the Structure of Knowledge Grounded Conversations

Behnam Hedayatnia
Amazon Alexa AI

Conversational agents like Amazon Alexa, Google Assistant and Apple Siri have been exploding in popularity over the past few years. However, much work remains in the area of social conversation over a broad range of domains and topics. To advance the state of the art in open domain dialog, Amazon launched the Alexa Prize, a 2.5-million-dollar university competition where selected university teams were challenged to build conversational agents, known as "socialbots", to converse coherently and engagingly with humans on popular topics such as Sports, Politics, Entertainment, Fashion and Technology for 20 minutes. The Alexa Prize offers a unique opportunity to perform research and interact with real user conversational data at scale. Over the past two years, we have learned that there are certain areas that these bots could improve on such as topical depth, breadth and smooth topical transitions in order to have deep and engaging conversations. Given this information, we formed a conversational dataset where we can study how to create engaging conversations. We introduce Topical-Chat: a knowledge-grounded human-human conversation dataset, where the underlying knowledge spans 8 broad topics. Our dataset enables models to leverage world knowledge while conversing with humans leading to more coherent and interesting conversations. We will present some modeling work using generative encoder-decoder conversational models trained on Topical-Chat and perform automated and human evaluation for benchmarking. Additionally we will present an analysis of Topical-Chat based on the knowledge content selected and presented to humans as background knowledge.

Maximizing Stylistic Control and Semantic Accuracy in NLG: Personality Variation and Discourse Contrast

Vrindavan Harrison, Lena Reed, Shereen Oraby, Marilyn Walker
Natural Language and Dialogue Systems Lab
University of California Santa Cruz
Santa Cruz, CA, US
{vharriso, lireed, soraby, mawalker}@ucsc.edu

Abstract

Neural generation methods for task-oriented dialogue typically generate from a meaning representation that is populated using a database of domain information, such as a table of data describing a restaurant. While earlier work focused solely on the semantic fidelity of outputs, recent work has started to explore methods for controlling the style of the generated text while simultaneously achieving semantic accuracy. Here we experiment with two stylistic benchmark tasks, generating language that exhibits variation in personality, and generating discourse contrast. We report a huge performance improvement in both stylistic control and semantic accuracy over the state of the art on both of these benchmarks. We test several different models and show that putting stylistic conditioning in the decoder and eliminating the semantic re-ranker used in earlier models results in more than 15 points higher BLEU for Personality, with a reduction of semantic error to near zero. We also report an improvement from .75 to .81 in controlling contrast and a reduction in semantic error from 16% to 2%.

1 Introduction

Neural encoder-decoder models were originally developed for machine translation (Sutskever et al., 2014; Bahdanau et al., 2014a), but they have also been shown to be successful in related natural language generation (NLG) tasks such as realizing dialogue system utterances from meaning representations (MRs) as shown for the restaurant domain in Table 1 (Dusek and Jurcícek, 2016). Recent work in neural NLG has shown that stylistic control is an important problem in its own right: it is needed to address a well-known limitation of such models, namely that they reduce the stylistic variation seen in the input, and thus produce outputs that tend to be dull and repetitive (Li et al., 2016). Here we compare different methods for directly controlling stylistic variation when generating from MRs, while simultaneously achieving high semantic accuracy.

Meaning Representation
name[Browns Cambridge], eatType[coffee shop], food[Italian], customerRating[average], area[riverside], familyFriendly[yes], near[Crowne Plaza Hotel]
E2E Dataset
Browns Cambridge is near Crowne Plaza Hotel. Browns Cambridge has a three star rating. Browns Cambridge is a family coffee shop.
Personality: Conscientious
Did you say Browns Cambridge? Well, i see, I think that it is a coffee shop, also it has a decent rating, and it is near Crowne Plaza Hotel kid friendly in riverside and an Italian place.
Personality: Disagreeable
Come on, I am not sure. I mean Browns Cambridge is an Italian place, also it has a damn average rating. It is near Crowne Plaza Hotel.
Personality: Unconscientious
Oh God I don't know! Browns Cambridge is a coffee shop, also it is family friendly near Crowne Plaza Hotel, also it is an Italian place in riverside, also it has like, a decent rating.

Table 1: Sample meaning representation with a realization from the E2E Challenge Dataset and three stylistic personality realizations.

Tables 1 and 2 illustrate the two stylistic benchmark datasets that form the basis of our experimental setup. Table 1 shows an example MR with three surface realizations: the E2E realization does not target a particular personality, while the other two examples vary stylistically according to linguistic profiles of personality type (Pennebaker and King, 1999; Furnham, 1990; Mairesse and Walker, 2011). Table 2 shows an example MR with two surface realizations that vary stylistically according to whether the discourse contrast relation is used (Nakatsu and White, 2006; Howcroft et al., 2013). Both of these benchmarks provide parallel data that supports experiments that hold constant the underlying meaning of an utterance, while varying the style of the output text. In

Proceedings of the 1st Workshop on Discourse Structure in Neural NLG, pages 1–12
Tokyo, Japan, November 1, 2019.

Meaning Representation
name[Brown's Cambridge], food[Italian], customerRating[3 out of 5], familyFriendly[no], price[moderate]
With Contrast Relation
Browns Cambridge is an Italian restaurant with average customer reviews and ***reasonable prices, but it is not child-friendly.***
Without Contrast Relation
Browns Cambridge serves Italian food in moderate price range. It is not kid friendly and the customer rating is 3 out of 5.

Table 2: A sample meaning representation with contrastive and non-contrastive surface realizations.

contrast, other tasks that have been used to explore methods for stylistic control such as machine translation or summarization (known as text-to-text generation tasks) do not allow for such a clean separation of meaning from style because the inputs are themselves surface forms.

We describe three methods of incorporating stylistic information as *side constraints* into an RNN encoder-decoder model, and test each method on both the personality and contrast stylistic benchmarks. We perform a detailed comparative analysis of the strengths and weaknesses of each method. We measure both semantic fidelity and stylistic accuracy and quantify the tradeoffs between them. We show that putting stylistic conditioning in the decoder, instead of in the encoder as in previous work, and eliminating the semantic re-ranker used in earlier models results in more than 15 points higher BLEU for Personality, with a reduction of semantic error to near zero. We also report an improvement from .75 to .81 in controlling contrast and a reduction in semantic error from 16% to 2%. To the best of our knowledge, no prior work has conducted a systematic comparison of these methods using such robust criteria specifically geared towards controllable stylistic variation. We delay a detailed review of prior work to Section 4 when we can compare it to our own.

2 Models and Variants

In the recent E2E NLG Challenge shared task, models were tasked with generating surface forms from structured meaning representations(Duek et al., 2019). The top performing models were all RNN encoder-decoder systems. Our model also follows a standard RNN Encoder–Decoder model (Sutskever et al., 2014; Bahdanau et al., 2014a) that maps a source sequence (the input MR) to a target sequence.

2.1 Model

Our model represents an MR as a sequence $x = (x_1, x_2, \ldots x_n)$ of slot-value pairs. The generator is tasked with generating a surface realization which is represented as a sequence y of tokens $y_1, y_2, \ldots y_m$. The generation system models the conditional probability $p(y|x)$ of generating the surface realization y from some meaning representation x. Thus, by predicting one word at a time, the conditional probability can be decomposed into the conditional probability of the next token in the output sequence:

$$p(y|x) = \prod_{t=1}^{m} p(y_t|y_1, y_2, \ldots y_{t-1}, x) \ . \quad (1)$$

We are interested in exercising greater control over the characteristics of the output sequence by incorporating *side constraints* into the model (Sennrich et al., 2016). The side constraints $\mathbf{c}$ act as an additional condition when predicting each token in the sequence. In this case, the conditional probability of the next token in the output sequence is given by:

$$p(y|x, \mathbf{c}) = \prod_{t=1}^{m} p(y_t|y_1, y_2, \ldots y_{t-1}, x, \mathbf{c}) \ . \quad (2)$$

In Section 2.2 we describe three methods of computing $p(y|x, \mathbf{c})$.

Encoder. The model reads in an MR as a sequence of slot-value pairs. Separate vocabularies for slot-types and slot values are calculated in a pre-processing step. Each slot type and slot value are encoded as one-hot vectors which are accessed through a table look-up operation at run-time. Each slot-value pair is encoded by first concatenating the slot type encoding with the encoding of its specified value. Then the slot-value pair is encoded with an RNN encoder. We use a multi-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to encode the input sequence of MR slot-value pairs. The hidden state $\bar{h_i}$ is represented as the concatenation of the forward state $\overrightarrow{h_i}$ and backward state $\overleftarrow{h_i}$. Specifically, $\bar{h_i} = (\overrightarrow{h_i}, \overleftarrow{h_i})$.

Decoder. The decoder is a uni-directional LSTM. Attention is implemented as in (Luong et al., 2015). We use a global attention where the attention scores between two vectors a and b are calculated as $a^T \mathbf{W} b$, where $\mathbf{W}$ is a model parameter learned during training.

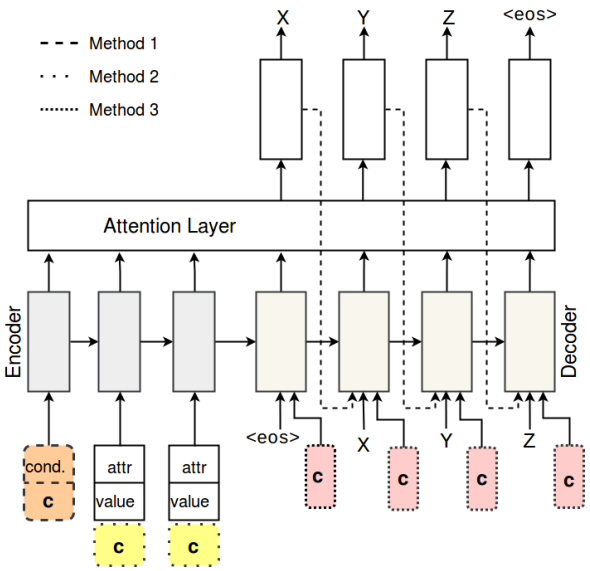

Figure 1: Attentional Encoder-Decoder architecture with each of the three side constraint implementations shown. The output sequence X, Y, Z is being generated from an MR represented as an input sequence of attribute value pairs.

2.2 Side Constraints

Recent work has begun to explore methods for stylistic control in neural language generation, but there has been no systematic attempt to contrast different methods on the same benchmark tasks and thereby gain a deeper understanding of which methods work best and why. Here, we compare and contrast three alternative methods for implementing side constraints in a standard encoder-decoder architecture. The first method involves adding slot-value pairs to the input MR, and the second involves extending the slot-value encoding through a concatenation operation. In the third method, side constraints are incorporated into the model by modifying the decoder inputs. The three side constraint implementation methods are shown simultaneously in Figure 1. The orange area refers to Method 1, the yellow areas corresponds to Method 2, and the red areas corresponds to Method 3.

Method 1: Token Supervision. This method provides the simplest way of encoding stylistic information by inserting an additional token that encodes the side constraint into the sequence of tokens that constitute the MR (Sennrich et al., 2016). We add a new slot type representing `side-constraint` to the vocabulary of slot-types, and new entries for each of the possible side

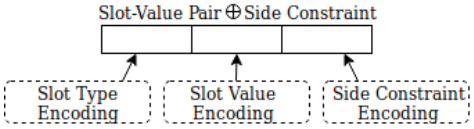

Figure 2: Slot-value encoding extended with constraint.

constraint values to the vocabulary of slot values.

Method 2: Token Features. This method incorporates side constraints through use of a slot-value pair feature. First we construct a vector representation c that contains the side constraint information. Normally the individual slot-value pair encodings are built by concatenating the slot-type with the slot-value as with Method 1. We modify each slot-value pair encoding of the MR by extending it with c, as seen in Figure 2.

Method 3: Decoder Conditioning. This method incorporates side constraint information into the generation process by adding additional inputs to the LSTM decoder. Traditionally, at the t-th time step a LSTM decoder takes two inputs. One input is the previous ground truth token's embedding w_{t-1}, and the other is a context vector d_t which is an attention-weighted average of the encoder hidden states. A vector c containing side constraint information is provided to the decoder as a third input. Thus at each time step the decoder's hidden state $\tilde{h}_i$ is calculated as

$$\tilde{h}_i = \text{LSTM}([w_{t-1}; d_t; c])\,. \tag{3}$$

3 Experiments: Varying Personality and Discourse Structure

We perform two sets of experiments using two stylistic benchmark datasets: one for personality, and one for discourse structure, i.e., contrast. In both cases, our aim is to generate stylized text from meaning representations (MRs). In the personality experiments, the generator's goal is to vary the personality style of the output and accurately realize the MR. The personality type is the side constraint that conditions model outputs, and is represented using a 1-hot encoding for the models that use side constraint Methods 2 and 3. For the sake of comparison, we also train a model that does not use conditioning (NoCon). In the discourse contrast experiments, the generator's goal is to control whether the output utterance uses the discourse contrast relation. The side constraint is

Personality	Realization
Meaning Representation	name[The Eagle], eatType[coffee shop], food[English], priceRange[cheap], customer rating[average], area[riverside], familyFriendly[yes], near[Burger King]
Agreeable	You want to know more about The Eagle? Yeah, ok it has an average rating, it is a coffee shop and it is an English restaurant in riverside, quite cheap near Burger King and family friendly.
Disagreeable	Oh god I mean, I thought everybody knew that The Eagle is cheap with an average rating, it's near Burger King, it is an English place, it is a coffee shop and The Eagle is in riverside, also it is family friendly.
Conscientious	I think that The Eagle is a coffee shop, it has an average rating and it is somewhat cheap in riverside and an English restaurant near Burger King. It is rather kid friendly.
Unconscientious	Yeah, I don't know. Mmhm ... The Eagle is a coffee shop, The Eagle is cheap, it's kind of in riverside, it is an English place and The Eagle has an average rating. It is kind of near Burger King.
Extravert	The Eagle is a coffee shop, you know, it is an English place, family friendly in riverside and cheap near Burger King and The Eagle has an average rating friend!

Table 3: Model outputs for each personality style for a fixed Meaning Representation (MR). The model was trained using control Method 3.

a simple boolean: contrast, or no contrast. The model is tasked with learning 1) which category of items can potentially be contrasted (e.g., *price* and *rating* can appear in a contrast relation but *name* can not), and 2) which values are appropriate to contrast (i.e., items with polar opposite values).

All models are implemented using PyTorch and OpenNMT-py[1](Klein et al., 2017). We use Dropout (Srivastava et al., 2014) of 0.1 between RNN layers. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and are optimized using stochastic gradient descent with mini-batches of size 128. Beam search with three beams is used during inference. We implement multiple models for each experiment using the methods for stylistic control discussed in Section 2.2. We tune model hyper-parameters on a development dataset and select the model of lowest perplexity to evaluate on a test dataset. All models are trained using lower-cased and delexicalized reference texts. The sample model outputs we present have been re-capitalized and re-lexicalized using a simple rule based script. Further details on model implementation, hyper-parameter tuning, and data processing are provided as supplementary material.

3.1 Benchmark Datasets and Experiments

Personality Benchmark. This dataset provides multiple reference outputs for each MR, where the style of the output varies by personality type (Oraby et al., 2018b).[2] The styles belong to the Big Five personality traits: agreeable, disagreeable, conscientious, un-conscientious, and extrovert, each with a stylistically distinct linguistic profile (Mairesse and Walker, 2010a; Furnham, 1990). Example model outputs for each personality on a fixed MR are in Table 3.

The dataset consists of 88,855 train examples and 1,390 test examples that are evenly distributed across the five personality types. Each example consists of a (MR, personality-label, reference-text) tuple. The dataset was created using the MRs from the E2E Dataset (Novikova et al., 2017) and reference texts synthesized by PERSONAGE (Mairesse and Walker, 2010b), a statistical language generator capable of generating utterances that vary in style according to psycho-linguistic models of personality. The statistical generator is configured using 36 binary parameters that target particular linguistic constructions associated with different personality types. These are split into *aggregation operations* that combine individual propositions into larger sentences, and *pragmatic markers* which typically modify some expression within a sentence, e.g. *tag questions* or *in-group markers*. A subset of these are illustrated in Table 4: see Oraby et al. (2018b) for more detail.

We conduct experiments using two control configurations that differ in the granularity of control that they provide. We call the first configuration *course-grained* control, and the model is conditioned using a single constraint: the personality label. The second configuration, called *fine-grained* control, conditions the model using the personality label and Personage's 36 binary control parameters as illustrated by Table 4, which provide fine-grained information on the desired style of the out-

[1] github.com/OpenNMT/OpenNMT-py

[2] nlds.soe.ucsc.edu/stylistic-variation-nlg

Attribute	Example
AGGREGATION OPERATIONS	
"WITH" CUE	*X is in Y, with Z.*
CONJUNCTION	*X is Y and it is Z. & X is Y, it is Z.*
"ALSO" CUE	*X has Y, also it has Z.*
PRAGMATIC MARKERS	
ACK_JUSTIFICATION	*I see, well*
ACK_YEAH	*yeah*
CONFIRMATION	*let's see ....., did you say X?*
DOWN_KIND_OF	*kind of*
DOWN_LIKE	*like*
EXCLAIM	*!*
GENERAL SOFTENER	*sort of, somewhat, quite, rather*
EMPHASIZER	*really, basically, actually, just*
TAG QUESTION	*alright?, you see? ok?*

Table 4: Example Aggregation and Pragmatic Operations

put text. The stylistic control parameters are not updated during training. When operating under fine-grained control, for side constraint Methods 2 and 3, the 1-hot vector that encodes personality are extended with dimensions for each of the 36 control parameters. For Method 1 we insert 36 tokens, one for each parameter, to the beginning of each input sequence, in addition to the single token that represents personality label.

Contrast Benchmark. This dataset provides reference outputs for 1000 MRs, where the style of the output varies by whether or not it uses the discourse contrast relation.[3] Contrast training set examples are shown in Table 2.

The contrast dataset is based on 15,000 examples from the E2E generation challenge, which consists of 2,919 contrastive examples and 12,079 examples without contrast.[4] We split the dataset into train and development subsets using a 90/10 split ratio. The test data is composed of a set of 500 MRs that contain attributes that can be contrasted, whose reference outputs use discourse-contrast (Reed et al., 2018). The test set also contains a set of 500 MRs that were selected from the E2E development set that do not use discourse-contrast. We crowd-sourced human-generated references for the contrastive test set, and used the references from the E2E dataset for the non-contrastive test set.[5]

[3] nlds.soe.ucsc.edu/sentence-planning-NLG

[4] www.macs.hw.ac.uk/InteractionLab/E2E/

[5] We will make our test and partitions of training data available to the research community if this paper is accepted.

3.2 Results

For both types of stylistic variation, we evaluate model outputs using automatic metrics targeting semantic quality, diversity of the outputs, and the type of stylistic variation the model is attempting to achieve. We also conduct two human evaluations. In the tables and discussion that follow, we refer to the models that employ each of the side constraint methods, e.g., Methods 1, 2, and 3, described in Section 2.2, using the monikers M{1,2,3}. The model denoted NoCon refers to a model that uses no side constraint information. Sample model outputs from the personality experiments are shown in Table 3. The outputs are from the M3 model when operating under the fine grained control setting. Outputs from model M2 of the contrast experiment are shown in Table 8.

3.2.1 Semantic Quality

Model	BLEU	SER	H	AGG	PRAG
Oraby et al. (2018b)					
NoCon	27.74	-	7.87	.56	.08
coarse	34.64	-	8.47	.64	.48
fine	37.66	-	8.58	.71	.55
This Work					
Train	-	-	9.34	-	-
NoCon	38.45	**0**	7.70	.44	.14
coarse control					
M1	49.04	0.000	8.49	.57	.51
M2	48.10	0.002	8.52	.62	.50
M3	49.06	0.009	8.50	.60	.50
fine control					
M1	55.30	0.004	8.77	.82	.94
M2	52.29	0.103	**8.80**	.84	.93
M3	**55.98**	0.014	8.74	.84	.93

Table 5: Automatic evaluation on Personality test set. *course* and *fine* refer to the specificity of the control configuration.

First, we measure general similarity between model outputs and gold standard reference texts using BLEU, calculated with the same evaluation script[6] as Oraby et al. (2018b). For the personality experiment, the scores for each conditioning method and each control granularity are shown in Table 5, along with the results reported by Oraby et al. (2018b). For the contrast experiment, the scores for each conditioning method are shown in Table 6, where we refer to the model and results of Reed et al. (2018) as *M-Reed*. Reed et al. (2018) do not report BLEU or Entropy (H) measures.

We first discuss the baselines from previous work on the same benchmarks. Interestingly, for

[6] github.com/tuetschek/e2e-metrics

Personality, our NoCon model gets a huge performance improvement of more than 11 points in BLEU (27.74 → 38.45) over results reported by Oraby et al. (2018a). We note that while the underlying architecture behind our experiments is similar to the baseline described by Oraby et al. (2018a), we experiment with different parameters and attention mechanisms. Reed et al. (2018) and Oraby et al. (2018b) also use an LSTM encoder-decoder model with attention, but they both implement their models using the TGen[7](Dušek and Jurcicek, 2016) framework with its default model architecture. TGen uses an early version of TensorFlow with different initialization methods, and dropout implementation. Moreover, we use a different one-hot encoding of slots and their values, and we implement attention as in Luong et al. (2015), whereas TGen uses Bahdanau et al. (2014b) attention by default. Side constraints are incorporated into the TGen models in two ways: 1) using a new dialogue act type to indicate the side constraints, and 2) a feed-forward layer processes the constraints and, during decoding, attention is computed over the encoder hidden states and the hidden state produced by the feed-forward layer. The TGen system uses beam-search and an additional output re-ranking module.

We now compare the performance of our own model results in Table 5. As would be expected, NoCon has the lowest performance overall of all models, with a BLEU of 38.45. With both coarse control and fine-grained control, M3 and M2 are the highest and lowest performers, respectively. For the contrast experiment, M2 and M3 have very similar values for all rows of Table 6. M2 has the highest BLEU score of 17.32 and M3 has 17.09. M1 is consistently outperformed by both M2 and M3. All side constraint models outperform NoCon. We note that the contrast task achieves much lower scores on BLEU. This maybe due to the relatively small number of contrast examples in the training set, but it is also possible that this indicates the large variety of ways that contrast can be expressed, rather than poor model performance. We show in a human evaluation in Section 3.2.2 that the contrast examples are fluent and stylistically interesting.

A comparison of our results versus those reported by Oraby et al. (2018b) are also shown in Table 5. Note that our model has an over 14 point margin of improvement in BLEU score when using coarse control and a more than 18 point improvement when using fine-grained control. Our models can clearly use the conditioning information more effectively than earlier work.

Model	BLEU	SER	H
Train	-		10.68
Contrast Data			
M-Reed	-	.16	-
NoCon	15.80	**.053**	**8.09**
M1	16.58	.055	8.08
M2	**17.32**	.058	8.03
M3	17.09	.058	7.93
Non Contrast Data			
NoCon	26.58	.025	7.67
M1	26.58	.023	7.56
M2	26.35	.017	7.68
M3	26.04	.035	7.40

Table 6: Automatic evaluation on Contrast test set.

Slot Error Rate. While the n-gram overlap metrics are able to measure general similarity between gold references and model outputs, they often do not do a good job at measuring semantic accuracy. Slot error rate (SER)(Wen et al., 2015; Reed et al., 2018) is a metric similar to word error rate that measures how close a given realization adheres to its MR. SER[8] is calculated using the slot aligner released[9] by Juraska et al. (2018) to count the number of attributes (slots) and their values that correctly (and incorrectly) occur in a given surface realization. Please refer to Supplementary Materials, Section A.1 for the definition of SER.

We evaluate each model using SER with results in Tables 5 and 6. We first note that all the SERs for both tasks are extremely low and that only M2 under fine control performs worse with an SER of .10. The models are clearly learning to realize the intended MRs. M1 has the best SER scores in all experiment conditions. In the contrast experiment, M2 and M3 are practically equivalent.

Model	Acc	Contrast Attempts
M-Reed	.75	422
M1	.74	437
M2	.79	485
M3	.81	474

Table 7: Contrast accuracy out of 500 examples.

[7] github.com/UFAL-DSG/tgen

[8] A formal definition of SER is provided in the supplementary materials.

[9] github.com/jjuraska/slug2slug

Meaning Representation	Realization
name[Aromi], eatType[restaurant], rating[low], familyFriendly[yes]	*Aromi is a **family friendly restaurant but the customer rating is low**.*
name[Fitzbillies], cuisine[English], price[more than $30], eatType[pub], familyFriendly[yes]	*Fitzbillies is a pub that serves English food, **is children friendly, but the price range is more than $30**.*
name[Clowns], price[more than $30], rating[high], familyFriendly[no], near[Clare Hall]	*Clowns is near Clare Hall. It has a **high customer rating but is not child friendly**.*
name[Cotto], cuisine[English], location[riverside], price[high], eatType[coffee shop], rating[5 out of 5], near[The Portland Arms]	*Cotto is a English coffee shop near The Portland Arms in the riverside. It has **a high price range but a customer rating of 5 out of 5**.*

Table 8: Sample outputs from model M2 with contrast relation in bold.

3.2.2 Quality in Variation

In the previous section we tested the ability of the side constraint models to produce semantically accurate outputs. In this section we evaluate the extent to which the side constraint models produce stylistically varied texts. We evaluate variation using two measures: 1) Entropy, and 2) counts on model outputs for particular stylistic targets.

Entropy. Our goal is NLG models that produce stylistically rich, diverse outputs, but we expect that variation in the training data will be averaged out during model training. We quantify the amount of variation in the training set, and also in the output references from the test set MRs using Entropy[10], H, where a larger entropy value indicates a larger amount of linguistic variation preserved in the test outputs.

The results are shown in the H column of Tables 5 and 6. For the personality experiment, the training corpus has 9.34 entropy and none of the models are able to match its variability. When using fine-grained control M2 does the best with 8.52 but all side constraint models are within 0.03. When using coarse control M2 has the highest entropy with 8.80. Our models with fine control outperform Oraby et al. (2018b) in terms of entropy. For the contrast experiment, NoCon has the highest entropy at 8.09, but the differences are small.

Counts of Stylistic Constructions. Entropy measures variation in the corpus as a whole, but we can also examine the model's ability to vary its outputs in agreement with the stylistic control parameters. Contrast accuracy measures the ratio of valid contrast realizations to the number of contrasts attempted by the model. We determine valid contrasts using the presence of polar opposite values in the MR and then inspecting realization of those values in the model output.

[10]A formal definition of our Entropy calculation is provided with the supplementary materials.

Table 7 shows the results. The row labeled M-Reed refers to the results reported by Reed et al. (2018). NoCon rarely attempts contrast because there is no way to motivate it to do so, and it therefore produces no contrast. Contrast attempts are out of 500 and M2 has the most at 485. In terms of contrast accuracy M3 is the best with 81%.

When comparing our model performance to M-Reed, models M{1,2,3} make more contrast attempts. M1 and M-Reed have similar contrast accuracy with 74% and 75%, respectively. The higher performance of our models is particularly impressive since the M-Reed models see roughly 7k contrast examples during training, which is twice the amount that our models see.

For personality, we examine each model's ability to vary its outputs in agreement with the stylistic control parameters by measuring correlations between model outputs and test reference texts in the use of the aggregation operations and pragmatic markers, two types of linguistic constructions illustrated in Table 4, and associated with each personality type. The results for these linguistic constructions over all personality types are shown in the last two columns (Agg, Prag) of Table 5. The supplementary material provides details for each personality. Our results demonstrate a very large increase in the correlation of these markers between model outputs and reference texts compared to previous work, and also further demonstates the benefits of fine-grained control, where we achieve correlations to the reference texts as high as .94 for pragmatic markers and as high as .84 for aggregation operations.

Methods Comparison. The results in Tables 5 and 7 reveal a general trend where model performance in terms of BLEU and entropy increases as more information is given to the model as side constraints. At the same time, the slot error rates are somewhat higher, indicating the difficulty of

simultaneously achieving both high semantic and stylistic fidelity. Our conclusion is that Method 3 performs the best at controlling text style, but only when it has access to a large training dataset, and Method 2 performs better in situations where training data is limited.

Human evaluation. We perform human evaluation of the quality of outputs for the M3 model with a random sample of 50 surface realizations for each personality, and 50 each for contrast and non-contrast outputs for a total of 350 examples. Three annotators on Mechanical Turk rate each output for both interestingness and fluency (accounting for both grammaticality and naturalness) using a 1-5 Likert scale.

Human evaluation results are shown in Table 9 for the personality experiment and Table 10 for contrast. The tables show average annotator rating in each category. For the personality outputs, each personality has similar fluency ratings with Conscientious slightly higher. The model outputs for the contrast relation have higher average ratings for Fluency than the non-contrastive realizations. For interestingness, we compare both the personality styles and the contrastive style to the basic style without contrast. The results show that non-contrast (3.07), the vanilla style, is judged as significantly less interesting than the personality styles (ranging from 3.39 to 3.51) or the use of discourse contrast (3.45) (p-values all less than .01).

	Con.	Dis.	Agr.	Ext.	Unc.	avg
Fluent	3.77	3.38	3.53	3.38	3.35	3.48
Interest	3.39	3.40	3.51	3.46	3.45	3.44

Table 9: Human evaluation results for personality.

	Non-contrast	Contrast
Fluent	4.21	4.38
Interest	3.07	3.45

Table 10: Human evaluation results for discourse contrast.

4 Related Work

Stylistic control is important as a way to address a well-known limitation of vanilla neural NLG models, namely that they reduce the stylistic variation seen in the input, and thus produce outputs that tend to be dull and repetitive (Li et al., 2016). The majority of other work on stylistic control has been done in a text-to-text setting where MRs and corpora with fixed meaning and varying style are not available (Fan et al., 2017; Iyyer et al., 2018; Wiseman et al., 2018; Ficler and Goldberg, 2017). Sometimes variation is evaluated in terms of model performance in some other task, such as machine translation or summarization. Herzig et al. (2017) also control personality in the context of text-2-text generation in customer care dialogues. Kikuchi et al. (2016) control output sequence length by adding a remaining-length encoding as extra input to the decoder. Sennrich et al. (2016) control linguistic honorifics in the target language by adding a special social formality token to the end of the source text. Hu et al. (2017) control sentiment and tense (past, present, future) in text2text generation of movie reviews. Ficler and Goldberg (2017) describe a conditioned language model that controls variation in the stylistic properties of generated movie reviews.

Our work builds directly on the approach and benchmark datasets of Reed et al. (2018) and Oraby et al. (2018b). Here we compare directly to the results of Oraby et al. (2018b), who were the first to show show that a sequence-to-sequence model can generate utterances from MRs that manifest a personality type. Reed et al. (2018) also develop a neural model for a controllable sentence planning task and run an experiment similar to our contrast experiment. Here, we experiment extensively with different control methods and present large performance improvements on both tasks.

5 Conclusion

We present three different models for stylistic control of an attentional encoder-decoder model that generates restaurant descriptions from structured semantic representations using two stylistic benchmark datasets: one for personality variation and the other for variation in discourse contrast. We show that the best models can simultaneously control the variation in style while maintaining semantic fidelity to a meaning representation. Our experiments suggest that overall, incorporating style information into the decoder performs best and we report a large performance improvement on both benchmark tasks, over a large range of metrics specifically designed to measure semantic fidelity along with stylistic variation. A human evaluation shows that the outputs of the best models are judged as fluent and coherent and that the stylistically controlled outputs are rated significantly more interesting than more vanilla outputs.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014a. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473 [cs, stat]*. ArXiv: 1409.0473.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014b. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jean-Marc Dewaele and Adrian Furnham. 1999. Extraversion: The unloved variable in applied linguistic research. *Language Learning*, 49(3):509–544.

Ondřej Dušek and Filip Jurcicek. 2016. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 185–190.

Ondrej Dusek and Filip Jurcícek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *CoRR*, abs/1606.05491.

Ondej Duek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *arXiv:1901.07931 [cs]*. ArXiv: 1901.07931.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv:1711.05217 [cs]*. ArXiv: 1711.05217.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, page 94104. Association for Computational Linguistics.

Adrian Furnham. 1990. Language and personality. In H. Giles and W. Robinson, editors, *Handbook of Language and Social Psychology*. Winley.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, page 249256.

Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 252–256.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

David M Howcroft, Crystal Nakatsu, and Michael White. 2013. Enhancing the expression of contrast in the sparky restaurant corpus. *ENLG 2013*, page 30.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1:18751885.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 13281338. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Penelope Levinson, Penelope Brown, Stephen C Levinson, and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

F. Mairesse and M.A. Walker. 2010a. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, pages 1–52.

François Mairesse and Marilyn A Walker. 2010b. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

Francois Mairesse and Marilyn A. Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Crystal Nakatsu and Michael White. 2006. Learning to say it well: Reranking realizations by predicted synthesis quality. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1113–1120.

Jekaterina Novikova, Ondřej Dušek, and 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*, pages 201–206, Saarbrcken, Germany. Association for Computational Linguistics.

Jon Oberlander and Alastair J Gill. 2004. Individual differences and implicit language: personality, parts-of-speech and pervasiveness. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Shereen Oraby, Lena Reed, TS Sharath, Shubhangi Tandon, and Marilyn Walker. 2018a. Neural multivoice models for expressing novel personalities in dialog. *Proc. Interspeech 2018*, pages 3057–3061.

Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018b. Controlling personality-based stylistic variation with neural natural language generators. In *SIGDIAL*.

J. W. Pennebaker and L. A. King. 1999. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77:1296–1312.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? *arXiv:1809.03015 [cs]*. ArXiv: 1809.03015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 3540. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):19291958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. *arXiv:1808.10122 [cs]*. ArXiv: 1808.10122.

A Supplementary Materials: Maximizing Stylistic Control and Semantic Accuracy in Dialogue Generation: Conditional Decoding for Personality Variation and Discourse Contrast

A.1 Calculating Slot Error Rate

Multiple methods of measuring SER have been proposed (Wen et al., 2015; Reed et al., 2018). In this work we use a method similar to the one described by Reed et al. (2018). First, we define the following types of errors: substitutions (realizing an attribute with an incorrect value), deletions (failing to mention an attribute), repeats, and hallucinations (mentioning an attribute that does not appear in the MR).

The SER score for a given (MR, text realization) pair is calculated by first calculating S, D, R, and $\tilde{H}$, which are the amounts of substitutions, deletions, repeats, and hallucinations, respectively. The SER formula is then given as:

$$\text{SER} = \frac{S + D + R + \tilde{H}}{N} \tag{4}$$

where N is the number of slots in the MR. Note that using this method can result in SER values greater than one, since it is possible for there to be more errors than slots in the MR.

A.2 Calculating Entropy

To calculate Shannon Text Entropy H, we first construct the corpus vocabulary V of all unigrams, bigrams, and trigrams. Then H is given by the equation

$$H = -\sum_{a \in V} \frac{k_a}{N} \cdot \log_2(\frac{k_a}{N}) \tag{5}$$

where N is the sum total of occurrences for all terms in V, and k_a is the number of occurrences for the term a.

A.3 Model Implementation Details

Model Implementation. All models are implemented using PyTorchand OpenNMT-py[11] (Klein et al., 2017). We use Dropout (Srivastava et al., 2014) of 0.1 between RNN layers. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and are optimized using stochastic gradient descent with mini-batches of size 128. Beam search with three beams is used during inference. We implement multiple models for each experiment using the methods for stylistic control discussed in Section 2.2. We tune model hyper-parameters on a development dataset and select the model of lowest perplexity to evaluate on a test dataset. All models are trained using lower-cased and de-lexicalized reference texts. The sample model outputs we present have been re-capitalized and re-lexicalized using a simple rule based script.

Hyper Parameter Tuning. Hyper parameters are tuned using a grid search over the following parameter space:

- RNN layers over the range [1, 2]
- RNN size over the range [150, 200, 250, 300]

We tune the number RNN layers and RNN size by training a model for each combination of layers and RNN size (8 models). We use the model of lowest development dataset perplexity to evaluate on the test dataset.

This parameter tuning process is performed for each of the side constraint methods and style parameter configuration (fine control, coarse control). The resulting hyper parameter values are shown in Table 11

Model	RNN layers	RNN size
NoCon	2	150
coarse control		
M1	1	200
M2	1	200
M3	2	150
fine control		
M1	1	200
M2	2	200
M3	1	200

Table 11: Model hyper-parameter values.

A.4 Data Processing

The data is pre-processed using Stanford CoreNLP (Manning et al., 2014).

A.5 Linguistic constructions: Pragmatic Markers and Aggregation Operations

Psycholinguistic studies have shown these markers to be indicative of the language of people with different personality traits (Pennebaker and King,

[11] github.com/OpenNMT/OpenNMT-py

1999; Furnham, 1990). For example, the use of pragmatic markers has been shown to effect perceptions of personality traits such as politeness, friendliness, extraversion, and enthusiasm (Oberlander and Gill, 2004; Levinson et al., 1987; Dewaele and Furnham, 1999). Using a method similar to Oraby et al. (2018b), we count the occurrences of pragmatic markers and aggregation operations in the model outputs. Then we average the counts within each personality category and calculate the Pearson correlation between the model output averages and the gold reference text averages.

The Pearson correlation r for pragmatic markers can be seen in Table 12. All values of r are significant with p-values less than 0.01. The model with no side constraints has $r \leq 0.17$ for all personalities except for conscientious with $r = 0.81$. This suggests that the un-constrained model picks one personality to optimize – conscientious in this case. For both control granularities each of the side constraint models have similar performance. Table 12 also shows the correlation results reported by Oraby et al. (2018b) where we observe a marked improvement in the pragmatic marker correlations of our models compared to theirs.

Pearson correlations for aggregation operations are shown in Table 13. Again, the test for correlation results in p-values less than 0.01 for each personality type. Here, the Token model of Oraby et al. (2018b) outperforms all three of our models when conditioning on only the personality label (coarse control).

Model	AGR	CON	DIS	EXT	UNC	avg
Oraby et al						
NoSup	0.05	0.59	-0.07	-0.06	-0.11	.08
Token	0.35	0.66	0.31	0.57	0.53	.48
Context	0.28	0.67	0.40	0.76	0.63	.55
This Work - coarse control						
NoCon	.17	.81	-.08	-.08	-.11	.14
M1	.44	.81	.17	.79	.32	.51
M2	.44	.81	.17	.83	.27	.50
M3	.40	.81	.14	.83	.31	.50
This Work - fine control						
M1	.87	.94	.98	.99	.90	.94
M2	.87	.94	.98	.99	.88	.93
M3	.87	.93	.97	.99	.90	.93

Table 12: Correlations between test examples and model outputs for pragmatic markers.

Model	AGR	CON	DIS	EXT	UNC	avg
Oraby et al						
NoSup	0.78	0.80	0.13	0.42	0.69	.56
Token	0.74	0.74	0.57	0.56	0.60	.64
Context	0.83	0.83	0.55	0.66	0.70	.71
This Work - coarse control						
NoCon	0.70	0.73	-0.19	0.35	0.60	.44
M1	0.67	0.70	0.58	0.56	0.36	.57
M2	0.61	0.70	0.58	0.60	0.60	.62
M3	0.64	0.68	0.58	0.59	0.49	.60
This Work - fine control						
M1	0.84	0.91	0.78	0.81	0.78	.82
M2	0.89	0.92	0.78	0.79	0.84	.84
M3	0.86	0.91	0.79	0.82	0.81	.84

Table 13: Correlations between test examples and model outputs for aggregation operations.

Incorporating Textual Evidence in Visual Storytelling

Tianyi Li **Sujian Li**
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
Peng Cheng Laboratory, Shenzhen, China
{litianyi01,lisujian}@pku.edu.cn

Abstract

Previous work on visual storytelling mainly focused on exploring image sequence as evidence for storytelling and neglected textual evidence for guiding story generation. Motivated by human storytelling process which recalls stories for familiar images, we exploit textual evidence from similar images to help generate coherent and meaningful stories. To pick the images which may provide textual experience, we propose a two-step ranking method based on image object recognition techniques. To utilize textual information, we design an extended Seq2Seq model with two-channel encoder and attention. Experiments on the VIST dataset show that our method outperforms state-of-the-art baseline models without heavy engineering.

1 Introduction

Multi-image visual storytelling is extended from a long trend of research in image captioning and has attracted considerable attention in recent years.

To generate the stories, previous work employed a Seq2Seq framework, using image encoder to encode the image sequences and sentence decoder to generate stories from encoded image sequences. Most of the researches (Smilevski et al., 2018; Kim et al., 2018; Gonzalez-Rico and Pineda, 2018; Wang et al., 2018b; Huang et al., 2018; Yu et al., 2017) focused on improving the decoder, and took simple concatenation or an LSTM as encoder. With such design, only images are utilized as input in generating the stories.

However, through our observations, the images alone are inadequate for visual storytelling. Storytelling is creative and diversified, so background knowledge is often required to convert a few images to a complete story. However, extracting such background knowledge is very difficult, especially with limited data.

To alleviate such drawback, it is important to take previous experience of story-writing into account. Imagining when a person starts to tell stories from images, he/she may not understand the implications in those images and fail to write a proper story. However, if he/she had heard others telling stories, he/she may be able to tell a story from the stories of similar image sequences he/she previously heard. Motivated by such process, we propose to utilize the large corpus as an inventory and improve the visual storytelling model by including stories from similar image sequences in corpus as input to strengthen the encoder design.

On building such models, two major problems need to be solved: (1) how to measure the relatedness of stories from the image sequence pair; (2) how to incorporate the textual information into the model so as to fully exploit it for storytelling.

To handle the first problem of picking the most relevant stories, we propose a two-step ranking method for their image sequences. We first filter out the 'dissimilar' images with object co-occurrence, and then sort the remaining candidates with feature vectors. For the second problem of incorporating textual information, we design an enhanced Seq2Seq model with two-channel encoder, one for visual input and the other for textual input.

We conduct experiments on the VIST dataset (Huang et al., 2016), a widely used multi-image visual storytelling dataset. We show that with textual evidence, our model outperforms our baselines and state-of-the-art models.

2 Method

Our method is based on the Seq2Seq framework, composed of a two-channel encoder and a RNN-based decoder. The whole architecture of our method is shown in Figure 1.

In the two-channel encoder, one channel en-

Proceedings of the 1st Workshop on Discourse Structure in Neural NLG, pages 13–17

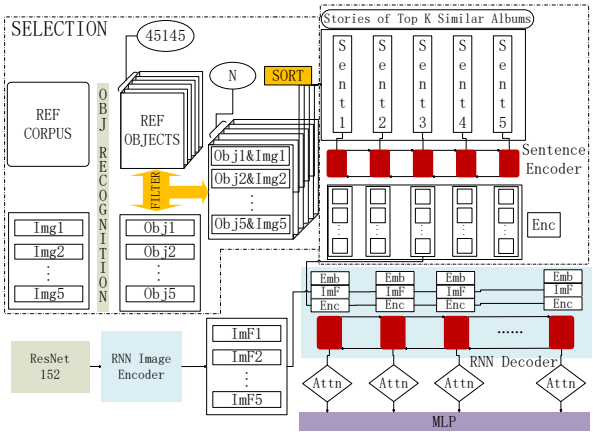

Figure 1: Overall architecture of our proposed method.

codes visual evidence from the image sequence and the other encodes textual evidence from relevant stories. In the decoder, we adopt another RNN model to generate stories from the two encoder outputs. To integrate the two types of information, we use Luong attention (2015) to dynamically attend to the stories. There are also other modifications, as further explained in 2.1.

To collect the textual evidence for encoder input, we design a selection method described in Section 2.2 to get stories from the most similar images.

2.1 Visual Storytelling Framework

Most previous works on visual storytelling followed the Seq2Seq framework, taking image recognition models such as ResNet (He et al., 2015) or Inception (Szegedy et al., 2016) to extract image features, feeding them into a story-level RNN encoder, bringing encoder output to the sentence-level decoder throughout the generation of the corresponding sentence.

We base our model on this framework with two key modifications: first, we design a text encoder to model the most similar stories which may provide evidence for story generation; second, we adopt the Luong attention Luong et al. (2015) mechanism on the textual side of encoded input to better utilize its information.

Text Encoder We use an RNN encoder to model the textual inputs. For each story, we feed its 5 sentences into the RNN one by one, retaining the hidden state across sentences. We take the RNN output of every step through the fully connected layers as encoder output.

Joint Decoder Different from previous methods, our decoder depend on both image and text encoder. The incorporation of the two encoders is the key problem. Here we adopt two approaches to solve this problem. First, we use the concatenation of the image encoder output, the embedding of last word and the last hidden states of sentence encoder as the input of the decoder. Second, we design a Luong attention layer in decoder to attend to sentence encoder outputs. Formally, the concatenation decoder can be denoted as:

$$s_t^i = DEC(s_{t-1}^i, [emb_{t-1}^i, sent^i_{len_{sent^i}}, img^i]) \quad (1)$$

and the downstream attention mechanism can be denoted as:

$$weights_t = s_t^i \cdot sent^i \quad (2)$$

$$C_t = Softmax(weight_t) \cdot sent^i \quad (3)$$

$$\pi_\beta(w_t^i | w_{1:t-1}^i) = softmax(W_c \cdot [C_t, s_t^i] + b_c) \quad (4)$$

where DEC is decoder RNN, s_t^i is RNN output for image i at step t, emb is word embedding, img and $sent$ are image and sentence encoder output, W_c and b_c are appended linear matrix and bias.

To be noticed, in our model, both decoder RNN and image encoder are generic and not limited to one particular design. The image encoder can be of arbitrary architecture as long as it generates a vector for each image, and the decoder RNN can also be designed flexibly as long as it takes a vector as input and outputs another vector at each step.

Specifically, we implemented these modifications on two popular systems: GLACNet (2018), the group with best human evaluation scores in Visual Storytelling Challenge NAACL 2018, wwho use residual encoder to generate GLOCAL vectors; XE-ss, a baseline model of Wang et al. (2018b), who proposed to improve performance with reinforcement model (AREL). We call our two models GLAC-TG and XE-TG. (see section 3.1 for details).

2.2 Textual Evidence Selection

To provide strong textual evidence for story generation, we aim to select stories which are most similar to the expected story for the given sequence of images.

With the assumption that similar images usually have similar stories, we take stories of similar im-

ages as similar stories. While it's most straightforward to choose the image with the most similar feature vector, it's shown through experiments 2 that comparing each pair of feature vectors for a large image corpus would be computationally expensive and suffer severely from false positives. Therefore, we propose to employ a two-step filter-and-sort method to pick out the most similar stories.

2.2.1 Filter

In the filter step, we use object co-occurrence to discriminate 'roughly similar' image sequences from 'dissimilar' ones. Here we filter by image object information because it conforms with the intuition that images with similar objects describe relevant events. It is also because object information has been widely used in image captioning as helpful information on images. (Mishra and Liwicki, 2019; Liu et al., 2018; Jiang et al., 2018; Anderson et al., 2017; Yin and Ordonez, 2017; Wang et al., 2018a).

We first get the types and numbers of objects in each image using an object recognition model, and then we measure image similarity with a categorical criterion and a numerical criterion. Formally, O_a and O_b are the set of objects present in image a and b respectively, c_x^k is the count of occurrence for object k in image x. The categorical criterion concerns the types of common objects, namely $score_{cat} = \frac{|O_a \cap O_b|}{\sqrt{|O_a||O_b|}}$; the numerical criterion concerns the differences in times of occurrence, namely $score_{num} = \frac{|O_a||O_b|}{|\Sigma_{k \in (O_a \cup O_b)}(c_a^k - c_b^k)^2|}$. Additionally, we set similarity scores to 0 when no objects are recognized in either image.

As mentioned above, we compare images in sequences. We measure the similarity between the sequences as the average score of its images. By filtering on the corpus and keeping only the image sequences scored on the top, we narrow down our candidate sequences to a modest size.

2.2.2 Sort

After obtaining a small set of roughly similar image sequences, we use feature vectors to rank similarity more precisely. Here we experiment on two approaches: a simple cosine similarity measure and a Bi-Linear model with Meteor score as gold annotation inspired by Cao et al. (2018). Empirically we find that Bi-Linear model shows no advantage against cosine similarity. Thus, we simply sort the roughly similar sequences with cosine similarity for downstream models.

3 Experiments

3.1 Experiment Setup

Our experiment is built on VIST (Huang et al., 2016) dataset, which is organized in 5-image sequences annotated with 5-sentence complete stories. The dataset size is 40098 for train, 4988 for validation and 5050 for test.

In GLAC-TG, we use LSTM RNN model with hidden size 1024, embedding size 256 and learning rate 1×10^{-3}; in XE-TG. We use GRU RNN model with hidden size 512, embedding size 512 and learning rate 4×10^{-4}.

In both models, we use ResNet152 (He et al., 2015) pre-trained on ImageNet (Krizhevsky et al., 2012) as image features, and we use Bi-LSTM and Bidirectional GRU respectively for image encoder.

In both models, we keep the hyper-parameters from their baseline models unmodified. For loss function, we use cross-entropy averaged on the sentence lengths.

On textual evidence selection, we use all stories and image sequences in train and validation set as reference corpus, and a Fast RCNN (He et al., 2017; Abdulla, 2017) model pre-trained on COCO dataset (Lin et al., 2014) to detect objects from each image. Roughly similar stories are filtered with numerical criterion at 500 candidate size as it shows the best performance.

3.2 Results

Methods	R	C	M
Huang et al. (2016)	-	-	31.4
Yu et al. (2017)	29.5	7.5	34.1
Gonzalez-Rico and Pineda (2018)	29.2	5.1	34.4
Huang et al. (2018)	30.8	10.7	35.2
GLACNet(2018) (re-trained)	26.3	2.2	33.0
GLAC-TG-top1(ours)	**26.5**	**2.0**	**33.4**
XE-ss(2018b)	29.7	8.7	34.8
AREL(2018b)	29.9	8.4	35.2
XE-TG-top1(ours)	**30.0**	**8.7**	**35.5**
XE-TG-top3(ours)	29.6	8.3	35.4
XE-TG-top1-attn(ours)	**29.9**	**9.2**	**35.2**
XE-TG-top3-attn(ours)	29.4	9.2	35.0
XE-TG-only	29.1	7.7	34.8

Table 1: Performance of our method compared to existing visual storytelling models, R is ROUGE-L, C is CIDEr, M is METEOR (models we re-trained in same setting as original are listed in (re-trained) rows)

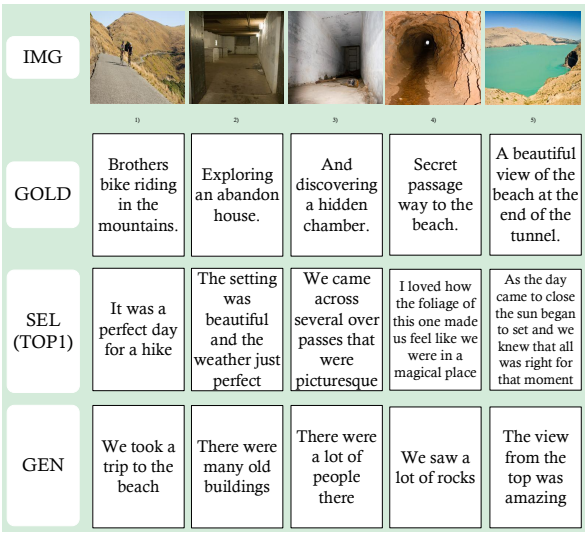

Figure 2: An example sequence of visual storytelling.

In Table 1, we compare our models with several strong baselines on three automatic evaluation metrics, ROUGE-L, CIDEr and METEOR. In the top block of Table 1, we present 4 previous baselines: 1) a standard Seq2Seq baseline model developed by Huang et al. (2016); 2) a hierarchically attentive model designed by Yu et al. (2017); 3) the Seq2Seq model with sentence-wise separate decoders by Gonzalez-Rico and Pineda (2018); 4) reinforcement learning with topic guided decoders by Huang et al. (2018). In the middle block, we present the GLACNet model Kim et al. (2018) and our improved GLAC-TG model. In the bottom block, we present our XE-TG models which are improved based on the XE-ss model in AREL framework (Wang et al., 2018b). For fair comparison, we evaluate all models with the open source evaluation code[1] (Yu et al., 2017).

Result shows that both our models outperform their corresponding baselines. Even using textual evidence only, our XE-TG-only model shows competitive performance compared to the baselines. Moreover, our XE-TG models using cross entropy loss outperformed state-of-the-art baselines with reinforcement learning techniques (Wang et al., 2018b; Huang et al., 2018). By using simple cross entropy loss, our models are also less costly to train, easier to tune and more stable when re-trained.

We conduct a qualitative analysis on XE-TG-top1 model in Figure 2 as an example. It shows that the selected similar story shares the same topic of wilderness adventure with similar story-flows. The generated story also catches the essence of the image sequence, with basic details closely relevant. It shows that our textual evidence selection method is capable of selecting proper textual evidence, and our storytelling framework is capable of capturing the provided information and telling fluent and coherent stories.

3.3 Analysis on Textual Evidence Selection

In this section, we further explore the effectiveness of similar stories. We experimented on filtering candidate size 50, 100 and 500 with both categorical and numerical criteria, using sorting on the entire reference corpus for comparison and METEOR score as a metric of actual story similarity. In Table 2, we show that for all methods, the selected stories are significantly more similar to gold stories than randomly selected ones, and stories with higher rankings are generally better than those with lower rankings. Moreover, for both criteria, candidate size poses negligible effect.

On the other hand, neither sorting on full corpus nor sorting by bi-linear model shows competitive results compared to our approach.

M	categorical			numerical		
	50	100	500	50	100	500
1	24.8	24.8	25.0	24.9	24.7	24.5
2	24.9	24.8	24.7	24.4	24.5	24.6
3	24.6	24.5	24.6	24.6	24.6	24.5
4	24.5	24.9	24.8	24.5	24.5	24.3
5	24.8	24.6	24.6	24.5	24.5	24.5
rand	23.8					
full	23.28 (average on top 5)					
B-L	23.62 (average on top 5)					

Table 2: METEOR scores for top 1 to 5 similar stories regarding two criteria, B-L refers to Bi-Linear

4 Conclusion

In this paper, we show that textual evidence from similar image sequences contains rich information for visual storytelling, therefore it's capable of boosting storytelling performance. We propose a feasible two-step approach to extract textual evidence from a large corpus. We also design a two-channel encoder to incorporate textual and visual evidence into the Seq2Seq visual storytelling models and achieve state-of-the-art performance with-

[1] https://github.com/lichengunc/vist_eval

out heavy engineering.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on this paper. This work was partially supported by National Natural Science Foundation of China (61572049 and 61876009).

References

Waleed Abdulla. 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and VQA. *CoRR*, abs/1707.07998.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia. Association for Computational Linguistics.

Diana Gonzalez-Rico and Gibran Fuentes Pineda. 2018. Contextualize, show and tell: A neural visual storyteller. *CoRR*, abs/1806.00738.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR*, abs/1703.06870.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Qiuyuan Huang, Zhe Gan, Asli Çelikyilmaz, Dapeng Oliver Wu, Jianfeng Wang, and Xiaodong He. 2018. Hierarchically structured reinforcement learning for topically coherent visual story generation. *CoRR*, abs/1805.08191.

Ting-Hao K. Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Aishwarya Agrawal, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*.

Wenhao Jiang, Lin Ma, Xinpeng Chen, Hanwang Zhang, and Wei Liu. 2018. Learning to guide decoding for image captioning. *CoRR*, abs/1804.00887.

Taehyeong Kim, Min-Oh Heo, Seonil Son, Kyoung-Wha Park, and Byoung-Tak Zhang. 2018. GLAC net: Glocal attention cascading networks for multi-image cued story generation. *CoRR*, abs/1805.10973.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Fenglin Liu, Xuancheng Ren, Yuanxin Liu, Houfeng Wang, and Xu Sun. 2018. simnet: Stepwise image-topic merging network for generating detailed and comprehensive image captions. *CoRR*, abs/1808.08732.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

Ashutosh Mishra and Marcus Liwicki. 2019. Using deep object features for image descriptions. *CoRR*, abs/1902.09969.

Marko Smilevski, Ilija Lalkovski, and Gjorgji Madjarov. 2018. Stories for images-in-sequence by using visual and narrative components. *CoRR*, abs/1805.05622.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Josiah Wang, Pranava Swaroop Madhyastha, and Lucia Specia. 2018a. Object counts! bringing explicit detections back into image captioning. *CoRR*, abs/1805.00314.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018b. No metrics are perfect: Adversarial reward learning for visual storytelling. *CoRR*, abs/1804.09160.

Xuwang Yin and Vicente Ordonez. 2017. OBJ2TEXT: generating visually descriptive language from object layouts. *CoRR*, abs/1707.07102.

Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2017. Hierarchically-attentive RNN for album summarization and storytelling. *CoRR*, abs/1708.02977.

Fine-Grained Control of Sentence Segmentation and Entity Positioning in Neural NLG

Kritika Mehta **Raheel Qader** **Cyril Labbé** **François Portet**
Univ. Grenoble Alpes, LIG
38000 Grenoble, France
kritika.mehta@grenoble-inp.org
raheel.qader@univ-grenoble-alpes.fr
{cyril.labbe,francois.portet}@imag.fr

Abstract

The move from pipeline Natural Language Generation (NLG) approaches to neural end-to-end approaches led to a loss of control in sentence planning operations owing to the conflation of intermediary micro-planning stages into a single model. Such control is highly necessary when the text should be tailored to respect some constraints such as which entity to be mentioned first, the entity position, the complexity of sentences, etc. In this paper, we introduce fine-grained control of sentence planning in neural data-to-text generation models at two levels - realization of input entities in desired sentences and realization of the input entities in the desired position among individual sentences. We show that by augmenting the input with explicit position identifiers, the neural model can achieve a great control over the output structure while keeping the naturalness of the generated text intact. Since sentence level metrics are not entirely suitable to evaluate this task, we used a metric specific to our task that accounts for the model's ability to achieve control. The results demonstrate that the position identifiers do constraint the neural model to respect the intended output structure which can be useful in a variety of domains that require the generated text to be in a certain structure.

1 Introduction

Typical NLG models are characterized by a pipeline of stages (Walker et al., 2007; Barzilay and Lapata, 2006; Walker et al., 2002; Stent, 2002; Barzilay and Lee, 2002; Langkilde and Knight, 1998; Reiter and Dale, 1997). This approach can be conceptually divided into solving two questions: *what to say?* aka content determination and planning, and *how to say it?* aka text realization (Gatt and Krahmer, 2018). In contrast, end-to-end NLG systems combine these stages in a single end-to-end learning framework. Recently, there has been a lot of interest in combining sentence planning and realization stage into a single neural model (Nayak et al., 2017; Dušek and Jurčíček, 2016; Lampouras and Vlachos, 2016; Wen et al., 2015; Mei et al., 2015). Although this resulted in some improvement at the grammatical level, in neural natural language generation this led to a loss of control that was otherwise possible in the pipeline approaches.

Neural NLG systems struggle to produce a consistent order of entities and are sometimes not faithful to the input by either hallucinating, omitting or repeating the entities (Moryossef et al., 2019). They do not allow control over the output structure and while they exhibit impressive levels of fluency, they are less equipped to deal with higher levels of text structuring in a consistent manner. They are also unable to generalize sentence planning operations beyond what is seen in the training. It is therefore important to introduce explicit control in neural NLG so that the output is faithful to the input. In this way, the system would be able to generate diverse realizations making way for explicit control over the output text structure.

By controlling the facts in the generated text, different variations can be produced that emphasize a particular fact which is more important than others. For example, if the focus should be on *"a cheap italian place"*, then *"There is a cheap italian place called The Sorrento. It is located in the city center."* will be more appropriate than *"The Sorrento is located in the city center. It is an Italian Restaurant. It is cheap too"*. This is particularly helpful in different domains, for instance, when generating hotel review summaries, it is important to put the elements important for the user in front (e.g., family, bathroom etc), when generating company descriptions, it is important to put the

Proceedings of the 1st Workshop on Discourse Structure in Neural NLG, pages 18–23

(1) **Alignment information**	"name[Blue Spice], eatType[coffee shop], area[city centre]", A coffee shop in the city centre area called Blue Spice.,(2: eattype) (21: area) (45: name)
(2) **Annotated reference text**	a (eattype)coffee shop in the (area)city centre area called (name)blue spice.
(3) **MR with sentence id.**	1_name[blue spice], 1_eattype[coffee shop], 1_area[city centre]
(4) **MR with sentence and slot id.**	1_3_name[blue spice], 1_1_eattype[coffee shop], 1_2_area[city centre]

Table 1: Example of an MR augmented with sentence and slot position identifiers.

main company selling points in front, and when generating messages for user with low literacy it is important to break sentences in small pieces, etc.

Recently there has been some work on controlling outputs of neural NLG models. In (Reed et al., 2018), authors use token supervision to reproduce sentence planning and discourse operations where a sentence scoping operation controls the number of sentences in the generated output which is measured using the period operator. This method does not provide any information about the word order in a particular sentence. While in (Moryossef et al., 2019) an explicit and symbolic text planner is proposed which determines the information structure and expresses it in the form of ordered trees. The plan structures in this work take the form of ad-hoc explorations for specific tasks and does not evolve into general-purpose plan structures. Moreover, this work is dataset dependent and does not generalize to datasets other than graph-based ones. To improve over the work in the literature of controlling neural NLG systems, in this paper, we propose an approach to explicitly control the realization of input entities in the desired sentences and in the desired position among individual sentences.

2 Overview of the Approach

Our method focuses on the control of sentence planning at two levels - 1) realization of input facts in the desired sentences and 2) realization of input facts in the desired position in the individual sentences. The idea is to directly attach sentence identifiers and slot identifiers to each slot, that indicate the sentence number and the position of the slot within that sentence respectively. The next step is to feed the modified Meaning Representation (MR) as an input to the seq2seq model and test if the model is able to learn to realize the slots in the correct positions.

2.1 Data Preparation

We used the E2E dataset for the experiments which provides information about restaurants and consists of about 50k combinations of a dialogue-act-based MR and 8.1 text references on an average. Each MR consists of up to 8 slots/attributes and their corresponding values.

As the dataset does not already contain a sentence plan, we modified it in a way that the MRs contain sentence and slot position identifiers. More specifically, given a reference text, two position identifiers were attached to each slot of the MR representing the sentence number in which the slot is found and the location of the slot in that specific sentence. The alignment information is extracted using a script[1] provided by the authors of Juraska et al. (2018). Table 1 provides an example of different stages of aligning the reference text with the MR. Initially, we annotate the reference text to identify the beginning of each slot value in the text (*line 2*), then using this annotation, we first attach the sentence identifiers (*line 3*), and finally the MRs are augmented with the position of each slot within a sentence (*line 4*). Owing to faults in the alignment information, in some cases the slot values are not detected in the reference text despite being present in the MR and for other cases the sentences do not contain some slots present in the MR. For such cases, a position token in the format 0_0_Slottype[Slot value] is attached.

Ideally, we expect a human to assign the position identifiers to each slot in the MR based on the desired output text. However, doing so for 4000 samples of the test set for validating our work would be very exhaustive. Therefore, we experiment with two different strategies to attach the position identifiers. Firstly, we directly use the test set reference text to extract the position identifiers. However, this will lead to biased results when computing the automatic metrics scores. Second

[1] https://github.com/jjuraska/slug2slug/tree/master/data/rest_e2e

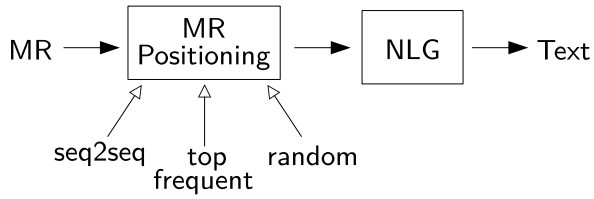

Figure 1: The three proposed approaches to automatically assign position identifiers to an MR.

strategy is to predict the position identifiers automatically. For this, we propose three different approaches as shown in Figure 1 and described in detail below:

- **Random:** The position identifiers are randomly chosen and assigned to the slots in the MR by taking care of a few rules.
- **Seq2seq:** The position identifiers are learned using a sequence-to-sequence (seq2seq) model. The train set MR without the position identifiers and the corresponding train set MR with the position identifiers are fed as training data to the model described in the Section 3. The test set MR without the position identifiers are then fed to the trained model, which outputs the test set MR with learned position identifiers.
- **Top Frequent:** The position identifiers for the test set are obtained from the most frequent combination of position identifiers in the train set. For each entry in the test set, the slot types (e.g., name) and slot values (e.g, blue spice) are separated and then the training entries which have slots with the same values are identified. Then, the most commonly occurring position identifier combination is picked for the test set entry.

Our random approach could be considered as a very naive baseline as randomly assigned identifiers might not even make sense in some cases. For example, having only one single slot in the first sentence (which never happens in the training set) would not be enough to form a grammatically correct sentence. Thus, in such cases, the model will fail to follow the assigned identifiers. The seq2seq model should perform much better than the random approach since it learns how to put position identifiers directly from the training data. However, this model has to re-generate the whole sequence including the slot type, slot value, and the newly added position identifiers. This means that any errors introduced during the generation process will significantly effect the text generated by the NLG model. Lastly, the top frequent approach is expected to perform better than the other two approaches. It extracts the most common sentence plan for some given values directly from the training set. Thus, the chosen sentence plan is among the ones that the NLG model was most exposed to during the training process.

2.2 Evaluation Metrics

In most NLG problems, sentence level metrics such as BLEU, ROUGE and METEOR are used to evaluate the results. However, these metrics are not entirely suitable to evaluate slot positioning since they measure performance at sentence level and do not provide precision in terms of sentence planning accuracy. In addition to using these automatic metrics to compare the results of the position augmented dataset with the original dataset, we evaluate the sentence planning accuracy using a modified version of the word error rate called the Slot Error Rate (SER) which is computed as:

$$SER = \frac{S + D + I}{N},$$

where S refers to number of substitutions, D refers to number of deletions, I refers to number of insertions and N refers to the total number of slots in the input MR. If a slot is realized in the wrong sentence it is counted as *wrong-sentence* substitution whereas if a slot appears in the wrong position but the correct sentence, it is counted as a case of *wrong-slot-position* substitution. It is important to note that the slot error rate compares two sequences of different nature (MR vs text). The generated text is pre-processed to extract the expressed slots and position of each slot using a script with some heuristics that involves lots of E2E dataset-specific handwritten rules. It is worth mentioning that our definition of SER is slightly different from the ones in the literature, particularly (Reed et al., 2018). In our case we take into account the exact position of each slot in the generated text, thus, the positioning of the realized slot in the generated text plays has a huge impact on the SER. Because of this difference, our metric can also be interpreted as slot position error rate in the realized text.

MR: 1_1_name[the cricketers], 1_2_eattype[restaurant], 1_3_food[chinese], 1_4_pricerange[20-25], 4_1_customer_rating[high], 2_1_area[riverside], 0_0_familyfriendly[no], 3_1_near[all bar one] **Output:** the cricketers is a restaurant providing chinese food in the 20-25 price range. it is located in the riverside. it is near all bar one. its customer rating is high.
MR: 1_1_name[the cricketers], 1_3_eattype[restaurant], 1_2_food[chinese], 1_7_pricerange[cheap], 1_6_customer_rating[5 out of 5], 1_5_area[city centre], 1_8_familyfriendly[yes], 1_4_near[all bar one] **Output** the cricketers is a chinese restaurant near all bar one in the city centre with a customer rating of 5 out of 5 and is cheap and family friendly.

Table 2: Output examples: position identifiers added from text references.

Experiment Type	SER
MR with sentence identifier	27%
MR with sentence and slot identifier	7%

Table 3: Results: SER on the test set prepared from test set alignment information.

	SER	BLEU	ROUGE	METEOR
Random	32%	0.22	0.39	0.34
Seq2Seq	5%	0.20	0.40	0.27
Top Frequent	0.7%	0.30	0.49	0.35

Table 4: Results: SER and sentence-level metrics on different versions of the test set with sentence and slot identifiers

3 Model Architecture

We use a standard seq2seq model with attention (Bahdanau et al., 2014; Luong et al., 2015). The seq2seq model consists of an encoder and a decoder based on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). The encoder reads the position augmented MR tokens one by one and feeds each token into an embedding layer and then to an LSTM layer. Finally the LSTM-based decoder takes the last hidden state from the encoder and starts generating output tokens one by one. Our decoder uses the dot attention mechanism as described in (Luong et al., 2015).

4 Experiments and Results

We begin by comparing the SER obtained using the test set with only sentence identifier and the test set with both sentence and slot identifiers. The position identifiers are attached based on the alignment information obtained from the test set. Table 3 shows that the SER for the model with both sentence and slot identifiers is significantly lower. This is probably because the model in this case has more information to learn from and the slot identifier proves to be very important. In order to analyze the results better, in Table 2 some of the output examples are shown. As it can be seen, even though both examples have quite complex combination of position identifiers, all the slots are realized in the correct position as indicated in the MR.

In the next part of the experiments, instead of using the reference text of the test set, we used 3 other approaches listed in Section 2.1 to attach the sentence and slot identifiers. The idea here is to i) not rely on the reference text, since it will bias the automatic metrics such as BLEU, ROUGE and METEOR and ii) to try more complex position identifier combinations and test the robustness of the model. Table 4 summarizes the BLEU, ROUGE, METEOR, and SER results obtained on the test set using the 3 approaches. It can be seen that the sentence-level metrics scores for the different versions of position augmented test set are quite variable. Top frequent seems to significantly outperform the other two approaches. This can be attributed to the fact that top frequent uses the most common sentence plans from the train set which are the easiest for the model to generate. This also shows that the test set and the training sets are extremely similar in their sentences' structures. Surprisingly the seq2seq approach performs significantly worse than the top frequent one. As described earlier, this is mainly because the seq2seq model introduces many errors in the slot types and values during the generation process which does not happen in the top frequent approach. These errors are also propagated to the NLG model, and hence, the performance is significantly impacted. It is important to note that these sentence-level metrics are computed on a single reference as opposed to the E2E challenge systems where multiple references where used, and as a matter of fact, their results there were much higher. The reason that we cannot use multiple references is because in the case of the random approach, each MR is changed and is assigned a unique set of identifiers, thus, distinguishing all of

the previously unique MRs. To make the results consistent, we report single reference scores for the other two approaches as well.

When it comes to SER, we can see that top frequent again achieves the lowest score of 0.7%. The seq2seq model was trained with the position augmented train set that mostly consists of text with just one sentence. As data with one sentence is relatively easier than the data with multiple sentences, the SER of 5% with seq2seq model based test set is justifiable. The test set prepared with random identifiers reports an SER of 32%, which can be explained by the fact that the identifiers are attached without significant rules, and hence, some of the MRs do not make a logical sense if realized as text. Nevertheless, the model still learns to produce logically and grammatically correct sentences.

To better assess the faithfulness of the model, we used human verification of the model's output. We randomly selected 50 generated outputs (from samples of Table 3, *line 2*) and 4 annotators manually annotated the MR to show deletion, insertion, substitution and hallucination of slots where hallucination refers to realization of a wrong *slot value*. The original MR is compared with the new annotated MR to obtain an SER. The different scores obtained were averaged and the final SER reported is 14.25%. This score is slightly higher than the 7% reported in Table 3 since human subjects were additionally annotating hallucinations too. Excluding hallucinations will lead to a similar score obtained using the SER metric. Thus we can say that the human verification scores is consistent with the scores obtained from the SER metric.

5 Conclusion

We presented an approach to explicitly control the output text structure by incorporating control at two levels of sentence planning- realization of input entities in desired sentences and in the desired position among individual sentences. We created a new data set with position identifiers designed specifically for controlling sentence planning operations and we investigated different ways of preparing such sentence plans. Our results show that the model learns from the extra position identifiers which provide the capability to control variation in the output and enables generalizing to unseen combinations without a significant loss of performance in terms of sentence-level metrics.

Acknowledgments

This project was partly funded by the IDEX Université Grenoble Alpes innovation grant (AI4I-2018-2019) and the Région Auvergne-Rhône-Alpes (AISUA-2018-2019).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Regina Barzilay and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366. Association for Computational Linguistics.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 164–171. Association for Computational Linguistics.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Juraj Juraska, Panagiotis Karagiannis, Kevin K Bowden, and Marilyn A Walker. 2018. Slug2slug: A deep ensemble model with slot alignment for sequence-to-sequence natural language generation.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1101–1112.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. *arXiv preprint arXiv:1904.03396*.

Neha Nayak, Dilek Hakkani-Tür, Marilyn A Walker, and Larry P Heck. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *INTERSPEECH*, pages 3339–3343.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? *arXiv preprint arXiv:1809.03015*.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Amanda J Stent. 2002. A conversation acts model for generating spoken dialogue contributions. *Computer Speech & Language*, 16(3-4):313–352.

Marilyn A Walker, Owen C Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3-4):409–433.

Marilyn A Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Zero-shot Chinese Discourse Dependency Parsing via Cross-lingual Mapping

Yi Cheng **Sujian Li**
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
Peng Cheng Laboratory, Shenzhen, China
{yicheng,lisujian}@pku.edu.cn

Abstract

Due to the absence of labeled data, discourse parsing still remains challenging in some languages. In this paper, we present a simple and efficient method to conduct zero-shot Chinese text-level dependency parsing by leveraging English discourse labeled data and parsing techniques. We first construct the Chinese-English mapping from the level of sentence and elementary discourse unit (EDU), and then exploit the parsing results of the corresponding English translations to obtain the discourse trees for the Chinese text. This method can automatically conduct Chinese discourse parsing, with no need of a large scale of Chinese labeled data.

1 Introduction

Discourse parsing aims to analyze the inner structure of texts, which is fundamental to many natural language processing applications, such as question answering and summarization. The construction of discourse corpora has promoted the development of discourse parsing techniques. In English, the widely-used discourse corpora include the Rhetorical Structure Theory Treebank (RST-DT) (Carlson et al., 2001) and Penn Discourse TreeBank (PDTB) (Prasad et al., 2008).

Recently, Li et al. (2014a) and Yoshida et al. (2014) proposed the discourse dependency structure (DDS). DDS directly links the EDUs, so it has fewer nodes and simpler structures compared to RST and PDTB. In addition, it can easily represent non-projective structures, while hierarchical structures need other complex mechanisms to do so. DDS is especially important for Chinese. Kang et al. (2019) analyzes almost all the existing Chinese discourse treebanks and concludes that DDS is the future direction due to its right balance between expressiveness and practicality. However, little research has been done on Chinese DDS. On one hand, there have been no such DDS treebanks in Chinese yet. Most of the existing Chinese discourse corpora follow PDTB-style or RST-style annotation (Zhou and Xue, 2012, 2015; Ming, 2008). Building a high-quality DDS corpus from scratch is labor-intensive and there are some conversion problems in transforming an existing corpus into DDS. On the other hand, a Chinese discourse parser needs to explore efficient features through trial and error based on the characteristics of Chinese. For the above reasons, Chinese text-level dependency parsing remains challenging.

To overcome these problems, we propose a simple and efficient method that conducts zero-shot Chinese discourse dependency parsing by exploiting the existing English discourse resources, with no need for Chinese training data. This is motivated by the observation of some Chinese-English parallel sentences such as the examples in Fig.1, whose dependency parsing trees are the same. It can be seen from the figure that the logical organization of a text is similar at the macro discourse level regardless of languages, in spite of lexical or grammatical differences.

Based on this observation, we employ machine translation (MT) and English discourse parsing techniques to parse a Chinese text. Our proposed method is simple but feasible, because English discourse dependency parsing has made progress, especially in parsing discourse tree structures (Liu and Lapata, 2017; Kim et al., 2017), and Chinese-to-English MT techniques are relatively mature (Nikolov et al., 2018; Hadiwinoto and Ng, 2018). Specifically, we first make use of MT techniques to translate a Chinese text into English and then adopt a transition-based English parser to analyze the translated text. Finally, we map this English parsing result to the Chinese text. During this process, some modifications are made to MT and the parsing result for performance improvement .

Proceedings of the 1st Workshop on Discourse Structure in Neural NLG, pages 24–29
Tokyo, Japan, November 1, 2019.

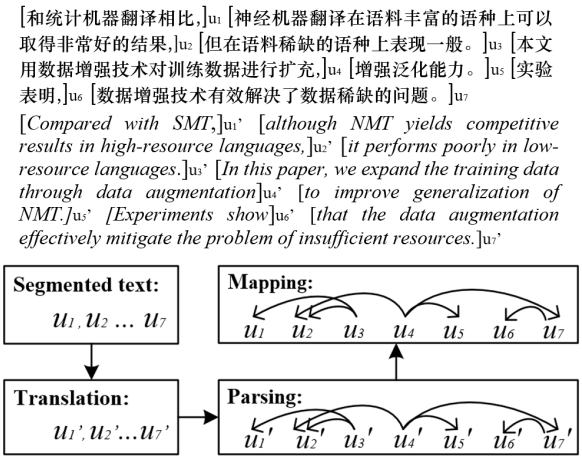

Figure 1: Illustration of Our Parsing Method via a Chinese-English Parallel Example

To evaluate our proposed method, we manually construct a small dataset, on which our method exhibits promising performance. This corpus will be released soon. The experiment results demonstrates that our method is potentially helpful in building large-scale data for Chinese neural NLG systems that make use of discourse structure. To the best of our knowledge, we are the first to conduct discourse dependency parsing in Chinese.

2 Chinese Discourse Dependency Corpus Construction

In this work, a small-scale Chinese discourse dependency treebank is constructed for evaluation. Here, we primarily follow the guideline of building the English discourse dependency treebank SciDTB (Yang and Li, 2018a) to explore the specifics of labeling DDS in Chinese.

First, scientific abstracts are chosen as corpus sources, because they are short texts with obvious logic and within the same domain as the English treebank (SciDTB) (Yang and Li, 2018a). Specifically, 108 abstracts are selected from a Chinese NLP journal *JCIP* [1].

Second, we manually separate these abstracts into elementary discourse units (EDUs), the basic units of a parsing tree. Each segmented abstract is checked at least twice to ensure segmentation quality. Our EDU segmentation mainly refer to the criteria of RST-DT (Carlson and Marcu, 2001) and make some modifications to the guideline based on the linguistic characteristics of Chinese (Cao et al., 2017; Yang and Li, 2018b). Due to space limitation, we do not list these modifications, as EDU segmentation is not the main work of this paper.

[1] http://jcip.cipsc.org.cn/CN/volumn/home.shtml

Relation	Frequency	Percentage/%
elab-addition	408	29.31
joint	236	16.95
enablement	138	9.91
bg-general	135	9.70
evaluation	85	6.10

Table 1: The Most Frequent Relation Types

Third, for each abstract, we identify the head of every EDU and the relation type between them, which is the most labor-intensive of all steps. We adopt the head and relation identification guidelines defined in Yang and Li (2018a). The relation categories include 17 coarse-grained and 26 fine-grained relation types. During the annotation process, some relation types are hard to distinguish (e.g., the distinction between the relations "manner-means" and "enablement" is vague). In addition, relation pronouns (e.g., that) and conjunctions (e.g., but) are used less frequently in Chinese (Li et al., 2014b), adding to the difficulty of relation labeling. The primary target of this study is to automate this step, i.e., to build the discourse tree with relation types between EDUs identified for a Chinese text.

Two annotators first learned the annotating principle before the annotation work. It takes the annotators 3 months to label the 108 abstracts, each being labeled at least twice independently in order to check annotation consistency and provide human performance as an upper bound. 30 abstracts are used for validation and the rest for test. The inter-annotator agreement is 0.780 and 0.673 with respect to UAS and LAS. In total, there are 1,500 EDUs (including 108 artificial root EDUs) with an average of 12.9 EDUs per abstract and 1,392 labeled discourse relations. On average, there are 2.91 EDUs per sentence and 22.17 characters per EDU. Table 1 shows the five most frequent relation types, along with their frequencies.

3 Zero-shot Chinese Dependency Parsing

As stated above, our method aims to generate a dependency parsing tree with relation types between EDUs identified for a Chinese text. It is assumed that golden EDU segmentation has al-

ready been conducted for the text. Formally, given a Chinese text t_C= $(u_1, u_2, ..., u_k)$ composed of k EDUs, we translate each Chinese EDU u_i directly into u'_i(i=$1, 2, ..k$), which can be seen as an English EDU. Translation performance is restrained to a certain extent because some EDUs cannot individually express their precise meaning when taken out of context. Thus, we make some modifications to the translation results before adopting a transition-based parser to generate a discourse dependency tree for the translated English EDUs. Finally, this dependency tree is mapped onto the EDU-segmented Chinese text. Fig.1 illustrates the whole process of our method. The main idea is simple. Only some technical issues in translation and text parsing need addressing, which will be introduced in the subsections.

3.1 Translation

We translate each Chinese EDU separately, instead of processing the whole text at a time, in order to obtain one-to-one correspondence between translated English EDUs and their Chinese counterparts , and to bypass EDU segmentation in English. But due to the absence of context information, translation accuracy is sacrificed, which degrades parsing performance. Since our work does not involve improving translation techniques, we only modify some obvious translation problems.

First, in translation, Chinese EDUs with incomplete meaning may be mistranslated into a sentence ended with a period. As Zhou and Xue (2015) point out, punctuation marks in Chinese can serve as clues of discourse relations. Most competitive Chiense discourse parsing models(Kang et al., 2016) use punctuation as one of their features. Therefore, we stipulate that the translated English EDUs can only be ended with a period if its corresponding Chinese EDU is ended with one. The other periods in the translation are replaced with commas.

Second, we modify the EDU identification of some relative pronouns because the position of them is helpful information for judging specific relation types (e.g.,"attribution"). Since we use EDUs as translation units , the EDU identification of some relation pronouns violates English EDU segmentation criteria. Take u_6 and u_7 in Fig.1 as example. Their translations are respectively: *[Experiments show that]*$\bar{u}'_6$, *[the data augmentation effectively mitigate the problem of insufficient resources.]*$\bar{u}'_7$. Our modification is to move "*that*" from $\bar{u}'_6$ to $\bar{u}'_7$, because a relative pronoun should be with the clause it introduces, according to the EDU segmentation criteria of RST-DT.

3.2 English Discourse Parsing

We follow the work of Yang and Li (2018a) and implement a two-stage transition-based dependency parser based on the idea of Wang et al. (2017) to conduct English parsing. In the first stage, the transition-based method for dependency parsing (Nivre, 2003) is adopted to identify the head for each EDU. We employ the action set of arc-standard system (Nivre et al., 2004), and an SVM classifier is designed to predict the most possible transition action. In the second stage, another SVM classifier is trained to predict relation types.

Since this parser is trained with SciDTB, its performance heavily relies on the features of the corpus. By analyzing the parsing results on the validation data, we find one obvious problem: the parser identifies the topic EDU (whose head is the root EDU, such as u_4 in Fig.1) with an accuracy of only 44.95%, while it reaches 85.06% on SciDTB.

To alleviate this problem, we first identify the topic sentence (which includes the topic EDU) in a rule-based way, because it usually begins with certain words, such as "该文"(this paper). Next, we split the passage into two parts with the topic sentence being the beginning of the latter part. The two parts are then parsed separately and joined together. In this way, the topic EDU identification accuracy increases to 68.52%.

4 Experiment

4.1 Setup

In our work, we compared several ready-made translation tools and chose to use *Youdao Translator*[2]. We referred to Yang and Li (2018a)'s work and implemented a two-stage transition-based discourse dependency parser to parse the English translated EDUs, with SciDTB as the training corpus. For comparison, we adopted the metrics of unlabeled and labeled attachment scores (UAS and LAS). UAS measures the accuracy of labeling the heads, while LAS measures the accuracy with respect to both head and relation labeling.

[2]http://fanyi.youdao.com/

	UAS	LAS
Random	0.188	0.013
Supervised(Chinese)	0.525	0.276
Zero-shot	**0.643**	**0.384**
Human(Chinese)	0.780	0.673
Supervised (English)	0.702	0.545
Human(English)	0.806	0.627

Table 2: Performance Comparison with Other Parsing Models

4.2 Results

Since there is no previous research on Chinese text-level dependency parsing, and our parsing approach is mainly designed to help construct a large-scale discourse dependency corpus in Chinese, our major concern is what performance this method (named *Zero-shot* in Table 2) can achieve and how it compares to human performance. We list several parsing results for comparison:

- *Random* is a transition-based dependency parser which randomly chooses "shift" or "reduce" as its next action and always uses the most frequent relation type "elab-addition" as the relation label. We test it on our Chinese corpus.
- *Supervised(Chinese)* is a two-stage transition-based dependency parser trained with 80 abstracts of our Chinese corpus and tested with the remaining 28 abstracts.
- *Supervised(English)* is a two-stage transition-based dependency parser trained on the training set of SciDTB and evaluated on its test set.
- *Human(Chinese)* and *Human(English)* are human performance on our Chinese discourse corpus and SciDTB respectively.

Table 2 shows the UAS and LAS of different parsing results. The top four rows are performance tested on our Chinese corpus and the bottom two on SciDTB. From *Human(English)* and *Human(Chinese)*, we can see that discourse labeling is a difficult task for both languages. Our *Zero-shot* method significantly outperforms the *Random* parser, meaning that parallel English and Chinese texts have similar discourse structures, and that our method effectively leverages such information. *Zero-shot* also performs about 12% and 11% higher than the *Supervised(Chinese)* with respect to the UAS and LAS metrics, because our corpus is too small to support supervision well

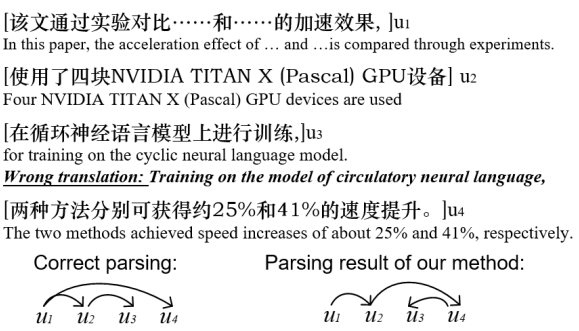

Figure 2: Parsing Errors Caused by Wrong Translation

Ablation test	UAS	LAS
Direct parsing	0.500	0.312
+ Relative Pronoun Adjustment	0.527	0.333
+ Punctuation Modification	0.607	0.353
+ Two-part Parsing	0.643	0.384

Table 3: Ablation Study

enough. Compared with *Supervised(English)*, the performance of *Zero-shot* is acceptable in terms of identifying the head EDU, but barely satisfactory in labeling the relations, which might be explained by different statistical distributions of relations types in Chinese and English..

To evaluate the contribution of each modification mentioned in Section 3, we conduct ablation experiments as shown in Table 3. The first line displays the performance of direct parsing without any modifications. The next three lines shows the performance with the modification strategies added in turn. As demonstrated in the table, these subtle modifications all play a useful role in improving performance.

Through error analysis, we find that many wrong cases can be corrected if the parser is given precise translation. Fig.2 provides an example where the heads of some EDUs are wrongly labeled, but are correct if given right translation. Translation precision can be improved with consideration of a larger context than EDU, which will be our future work.

5 Conclusions

In this paper, we present a simple and efficient method to conduct zero-shot Chinese discourse parsing, whose performance is close to the one of the state-of-art English parsers. It opens the possibilities for conducting dependency parsing on low-resource languages via cross-lingual mapping, re-

ducing human labor of corpus construction. In the future, we will further improve our method and test it in more languages and more domains.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on this paper. This work was partially supported by National Natural Science Foundation of China (61572049 and 61876009).

References

Shuyuan Cao, Nianwen Xue, Iria da Cunha, Mikel Iruskieta, and Chuan Wang. 2017. Discourse segmentation for building a rst chinese treebank. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 73–81. Association for Computational Linguistics.

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. Technical Report ISI-TR-545, University of Southern California Information Sciences Institute.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the SIGDIAL 2001 Workshop, The 2nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Saturday, September 1, 2001 to Sunday, September 2, 2001, Aalborg, Denmark*.

Christian Hadiwinoto and Hwee Tou Ng. 2018. Upping the ante: Towards a better benchmark for chinese-to-english machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Xiaomian Kang, Haoran Li, Long Zhou, Jiajun Zhang, and Chengqing Zong. 2016. An end-to-end chinese discourse parser with adaptation to explicit and non-explicit relation recognition. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning: Shared Task, CoNLL 2016, Berlin, Germany, August 7-12, 2016*, pages 27–32.

Xiaomian Kang, Chengqing Zong, and Nianwen Xue. 2019. A survey of discourse representations for chinese discourse annotation. *ACM Trans. Asian & Low-Resource Lang. Inf. Process.*, 18(3):26:1–26:25.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014a. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 25–35.

Yancui Li, Wenhe Feng, Jing Sun, Fang Kong, and Guodong Zhou. 2014b. Building chinese discourse corpus with connective-driven dependency tree structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 2105–2114.

Yang Liu and Mirella Lapata. 2017. Learning structured text representations. *CoRR*, abs/1705.09207.

Yue Ming. 2008. Rhetorical structure annotation of chinese news commentaries. *Journal of Chinese Information Processing*, 22:19–23.

Nikola I. Nikolov, Yuhuang Hu, Mi Xue Tan, and Richard H. R. Hahnloser. 2018. Character-level chinese-english translation through ASCII encoding. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 10–16.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural anguage Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 444–449. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Meeting of the Association for Computational Linguistics*.

An Yang and Sujian Li. 2018a. Scidtb: Discourse dependency treebank for scientific abstracts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 444–449. Association for Computational Linguistics.

Jingfeng Yang and Sujian Li. 2018b. Chinese discourse segmentation using bilingual discourse commonality. In *https://arxiv.org/abs/1809.01497*. arXiv.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1834–1839.

Yuping Zhou and Nianwen Xue. 2012. Pdtb-style discourse annotation of chinese text. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 69–77.

Yuping Zhou and Nianwen Xue. 2015. The chinese discourse treebank: a chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49(2):397–431.

Author Index