Workshop on Discourse Theories for Text Planning

Text or document planning as the mechanism of ordering messages in a coherent way for achieving a cohesive text has traditionally been realized by schemas or the establishment of rhetorical relations between messages and message sequences. Inspired by the descriptions of a comprehensive set of rhetorical relations in Rhetorical Structure Theory (RST), these relations have often been realized as planning operators for achieving a complete text. The rise of machine learning approaches for NLG tasks seems to hide the fact that RST-oriented text planning is by far not the only method for achieving linguistically relevant text plans.

Formal semantics and pragmatics offer a number of different theories on text organization and coherence phenomena whose explanatory power goes beyond the justified grouping of informational units, among them (Segmented) Discourse Representation Theory (SDRT), Question-under-discussion (QUD) approaches, and probabilistic approaches to meaning. This workshop addresses the relevance of these theories for text planning.

For example, QUD approaches to text structuring provide expressive theories facilitating concise analyses of a group of different pragmatic phenomena, ranging from the analysis of focus/background structures to dialogue moves, but they did not receive much attention for text planning issues in NLG. QUDs are the central concept in analyses that explain linguistic regularities as a consequence of the assumption that the sentences and text segments with which the regularities are associated are answers to an explicitly or implicitly asked question. QUDs were early on used for explaining possible sequences of dialogue moves (Carlson, 1983; Ginzburg, 1996), clarifying information structural concepts (e.g. the topic/focus distinction), temporal progression and foreground–background relations in narration, information structural constraints on implicatures (van Kuppevelt, 1996), representing discourse goals and defining contextual relevance (Roberts, 1996), and for analysing structure and coherence of discourse, of both text and dialogue (van Kuppevelt, 1995). Since then, QUDs have been firmly established as an analytic tool, leading to fruitful applications for a wide range of linguistic phenomena.

The aim of this workshop is to explore the interplay of linguistic theories of text planning-related phenomena with computational approaches to text planning,

be it rule-based or learning approaches, in order to bring these fields of expertise together.

Invited Speakers:

Jonathan Ginzburg (Université de Paris, CNRS)

Program Committee:

Nicholas Asher (IRIT Toulouse)
Anton Benz (ZAS Berlin)
Kees van Deemter (Utrecht University)
Claire Gardent (CNRS / LORIA Nancy)
Christoph Hesse (ZAS Berlin)
Ralf Klabunde (Ruhr University Bochum)
Maurice Langner (Ruhr University Bochum)
Tatjana Scheffler (Potsdam University)

Workshop Program:

The workshop took place on Tuesday December 15, 2020.

12:00-12:30	*Monological Text from Dialogue?* Jonathan Ginzburg
12:30-13:00	*Automatic planning of the dialogue between human and machine using discourse trees* Boris Galitsky & Dmitry Ilvovsky
13:00-13:30	*Neural Micro-Planning for Data to Text Generation Produces more Cohesive Text* Roy Eisentadt & Michael Elhadad
13:30-13:50	break
13:50-14:20	*Annotating QUDs for generating pragmatically rich texts* Christoph Hesse, Anton Benz, Maurice Langner, Felix Theodor, Ralf Klabunde
14:20-14:50	*Towards Domain-Independent Text Structuring* Grigorii Guz & Giuseppe Carenini

Automatic planning of the dialogue between human and machine using discourse trees

Boris Galitsky **Dmitry Ilvovsky**

Oracle National Research University Higher School of Economics

In many task-oriented chatbot domains, an objective is to fully inform a user about a particular important piece of information. It is also crucial to make user believe this piece of information, relying on explanation and argumentation in as much degree as possible. In some cases, it is important to make a user believe in a particular short text. This should be done by thoroughly navigating a user through possible disagreements and misunderstanding, to make sure the user is being explained and communicated an issue exhaustively.

Rather than throwing the whole paragraph of text at a user, we split it into logical parts and feed the user text fragment by fragment, following her interests and intents. To systematically implement this navigation, we follow a discourse-level structure for how the author of this text organized his thoughts. This can be done by navigating a discourse tree (DT) of this text. DT is a tree that is a labeled tree in which the leaves of the tree correspond to contiguous units for clauses (elementary discourse units, EDUs). Adjacent EDUs, as well as higher-level (larger) discourse units, are organized in a hierarchy by rhetorical relation (e.g., *Reason, Temporal sequence)* provided by Rhetorical structure theory (RST, Mann and Thompson, 1988). An anti-symmetric relation involves a pair of EDUs: nuclei, which are core parts of the relation, and satellites, which are the supportive parts of the rhetorical relation. A satellite can be delivered by the chatbot to a user as an utterance only if its nucleus has already been received and acknowledged in one way or another.

We outline the chatbot algorithm of the DT traversal, covering a multitude of user intents at each iteration:

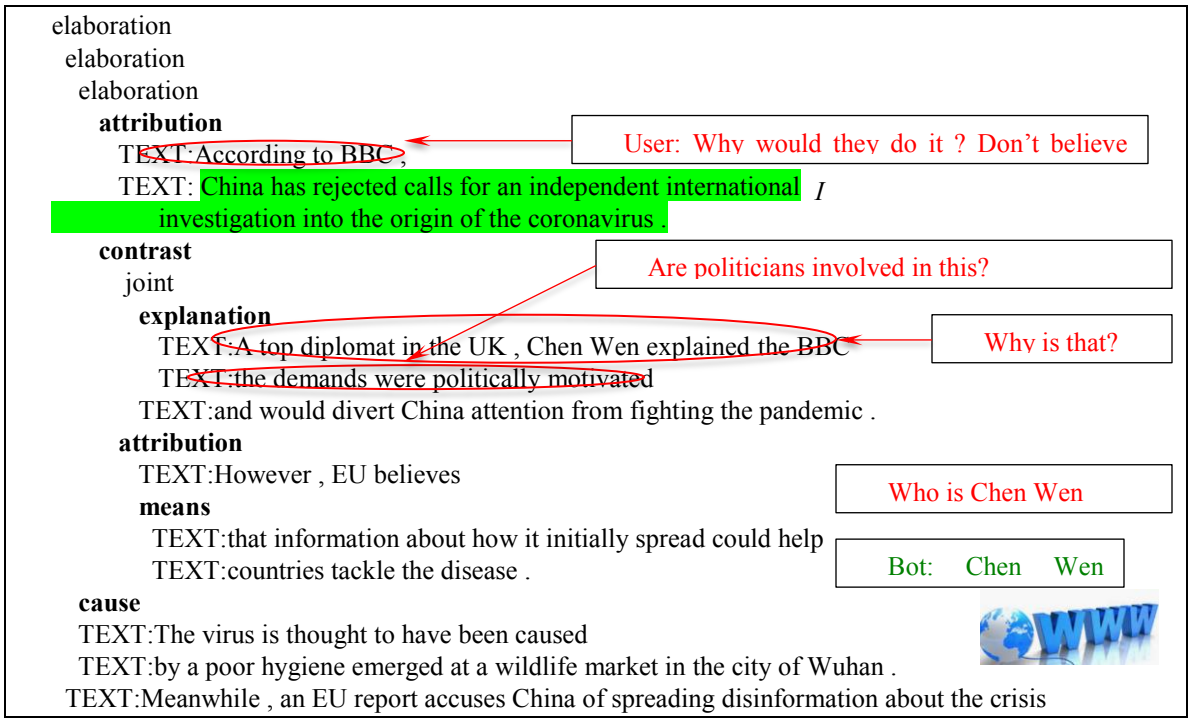
1. If a text is given, navigating a discourse tree of this text T is one of the most efficient ways to communicate it. The chatbot starts with making an introduction and then making the main statement M_T. Then the user would ask for more details E_T, disagree E_T or ask a question on a topic outside of the scope of this text O_T.
2. If the user asks for more details I_T, the EDU connected with *Elaboration* with M_T is provided as a reply. We denote this EDU as *Elaboration*(I_T). This is the easiest, most direct situation.
3. If the user disagrees, chatbot tries to find an EDU which is connected by Explanation or Cause with M_T or I_T. This EDU should be returned as a reply.
4. If the user asks a different question O_T then it should be answered as a factoid question but nevertheless the chatbot needs to take the user back to T so the reply should end with *Elaboration*(I_T).
5. If the user doubts about the validity of a claim in M_T, the chatbot needs to deliver *Attribution*(M_T) as an answer.

The procedure above should iterate till no more EDU in T is left or the user terminates the conversation. If the chatbot persistence is too high in trying to take the user back to T, this user would terminate the conversation too soon. Otherwise, if the chatbot persistence is too low, the user would deviate from T too far so will red less content of T (EDU(T)). We want to optimize the chatbot to maintain the optimal persistence to maximize the number of delivered EDU(T) till the conversation is abandoned by the user.

Let us take a text and show how a DT navigation leads a dialogue wrapped around this text.

According to BBC, China has rejected calls for an independent international investigation into the origin of the coronavirus. A top diplomat in the UK, Chen Wen explained the BBC the demands were politically motivated and would divert China attention from fighting the pandemic. However, EU believes that information about how it initially spread could help countries tackle the disease. The virus is thought to have been caused by a poor hygiene emerged at a wildlife market in the city of Wuhan.

A discourse tree for this text and a fragment of a sample navigation path is shown below.

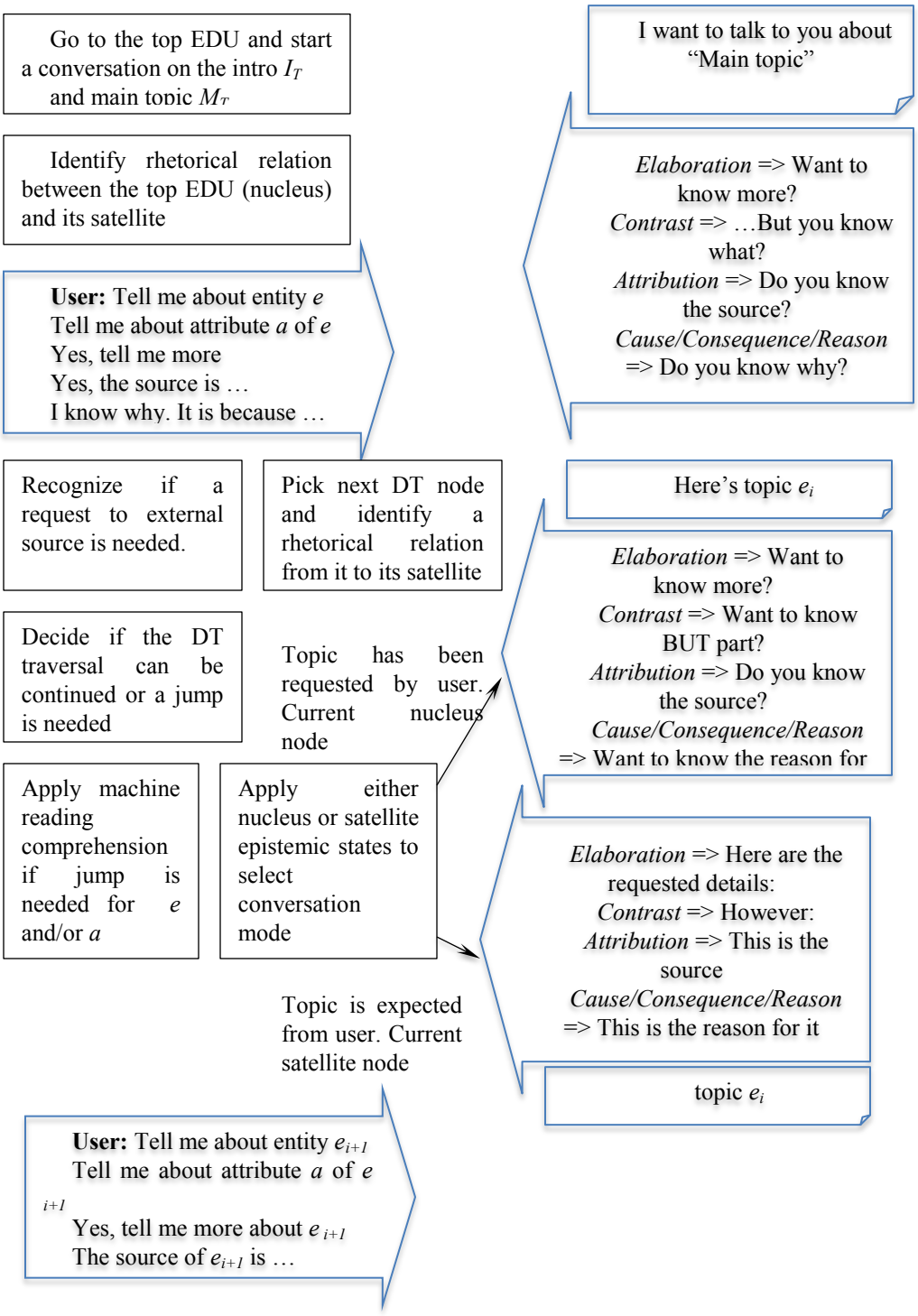


What we observe is that a dialogue is fairly plausible, although no data-driven method has been applied. It does not matter where the user deviates from the target text as long as the chatbot always takes her back to the EDU and rhetorical relation which is either relevant to what the user asked or claimed, or just follows the DT navigation flow from more important, closer to the root, to less important. If the user is asked a question outside of the scope of the target text, we provide an answer from the foreign source and then also switch topic and come back to the business of the target answer, proceeding with the DT navigation.

The dialogue flow based on navigation of a DT is shown below. A conversation with the focus on T starts with an Introduction of T followed by the main topic of T expressed by EDUs located closest to the root of DT. Chatbot utterance includes the information from the EDU of the current node plus an encouragement to the user to continue conversation, such as a question or a knowledge sharing request (on the top-right). Chatbot encouragement depends on the rhetorical relation for the current navigation node (now, the M_T node). The user replies (formulates a question) in a certain form, depending on the encouragement question of the chatbot (on the left).

User's question varies in terms of the focus entity or its attribute, and/or the epistemic state initiated by the chatbot. Once the user question is received by the chatbot, it is analyzed with respect to if an external knowledge source needs to be searched and/or if an Machine Reading Comprehension (MRC) component needs to be initiated to find a value for a factoid question and also identify an EDU this value occurs in. Then the decision needs to be made if the user changed the topic and a jump is required, or the chatbot can maintain the dialogue by continuing the DT navigation. Next navigation step depends on whether the current node is nucleus (and satellite is the next to be visited), or it is a satellite and its nucleus needs a visit. Epistemic state update is chosen accordingly.

For the nucleus, the user has already expressed his interest in a given topic. So information from its EDU is ready to be sent to the user. For the satellite, the user is encouraged to express his interest according to the rhetorical relation to the nucleus of this satellite. A topic is expected from this user. External search and/or MRC can be applied in this option.



References

Mann, W. and Thompson, S., Rhetorical Structure Theory: Toward a functional theory of text organization, 1988

Neural Micro-Planning for Data to Text Generation Produces more Cohesive Text

Roy Eisentadt, Michael Elhadad
Dept. of Computer Science, Ben-Gurion University of the Negev
Beer Sheva, Israel
royeis@post.bgu.ac.il, elhadad@cs.bgu.ac.il

Abstract

We aim to prove the usefulness of separating data to text generation into micro-planning and realization, and focus on micro-planning as a task that can be learned and evaluated separately. We adopt a simple structure for micro-plans and develop an initial neural model to learn such a plan from a flat input of triplets. We define a method to measure planning quality, and an evaluation method of generated text by examining syntactic phenomena related to text cohesion. In experiments on the WebNLG dataset, we demonstrate the correlation between higher quality planning and more natural, cohesive text. The quantitative data-driven methodological approach we illustrate can help formulate hypotheses that a more sophisticated micro-plan formalism and its interface with surface realization decisions could help explore.

1 Introduction

Traditional NLG pipelines distinguish distinct sub-tasks addressed by a generation system, including content determination, text structuring, sentence aggregation, lexicalization and surface realization (Gatt and Krahmer, 2018). Recent work on neural NLG has blurred the distinction among these sub-tasks and encouraged data-driven end-to-end approaches, such as transformer-based encoder-decoder architectures (Lewis et al., 2020). Recent data to text generation approaches are revisiting this decision, and show the benefit of dividing the full task into two steps: planning and realization (Moryossef et al., 2019; Castro Ferreira et al., 2019). The goal of micro-planning is to organize the input raw data into an interpretable and coherent information structure. Realization is then applied on this structure to generate coherent text that covers all the expected content without redundancy and without introducing unintended content. Planning and realization deal with distinct but closely related aspects of text structuring: planning is related to concepts from discourse theory such as rhetorical structure, information flow and coherence while realization handles the lexical and syntactic aspects of these concepts including information packaging, clause structuring and aggregation.

A modular approach brings two benefits: more control over each component of the generation and simpler modeling of both sub-problems when compared to end to end models. We hypothesize that an explicit planning model, including quality evaluation of plans, can lead to better control of the generated text in data-to-text tasks, and boost performance, as was indeed demonstrated in (Moryossef et al., 2019). We revisit this modularity argument with three new directions: (1) we study the extent to which a robust learned planning module can be derived (as opposed to a rule-based planning method); (2) we investigate whether an independent planning quality metric can be established, and the extent to which it correlates with end to end text quality metrics; (3) finally, we investigate specific aspects in realization that are directly related to micro-planning and cohesion and the extent to which good plans control their usage.

Applying learning methods to solve the task of planning is difficult for two main reasons: (1) available datasets (Gardent et al., 2017a,b) do not reward variability in plans. They contain a few pairs (data, text) for a given input (usually 3 to 5 variants per input), but there is no incentive to demonstrate a variety of plans to realize the same input; an ideal dataset to learn planning would instead hold different paraphrases for each entry based on changes in micro-planning; (2) Given a target text to generate, micro-plannings are not observable. One can come up with methods to derive a plan from a given text but the nature of the plan, how it is related to

observable syntactic structure is essentially an internal decision. Despite these obstacles, we aim to demonstrate the usefulness of learning an intermediate plan representation for data to text generation. Beyond this quantitative architectural analysis, the experimental setting we investigate provides a useful platform to explore more sophisticated models of text planning.

2 Datasets

Our experiments are based on the WebNLG 2020 dataset (Gardent et al., 2017a) which provides knowledge-graph to text entries. Each knowledge graph is composed of a set of logical forms that are encoded as triplets: (sub, rel, obj). For the purpose of learning and evaluation of the planning task, we utilize the DeepNLG dataset (Castro Ferreira et al., 2019), which is based on WebNLG 2017 and associates, for each (data, text) pair a manually derived plan, which has the structure of an ordered sequence of groups of triplets (one per observed sentence in the text). This data allows us to train in a supervised manner on the data-to-plan task.

3 Modeling Plans

We hypothesize that the task of generating text from a set of triplets $\mathcal{T} = t_1, \ldots, t_n$ will be improved if we model it as a pipeline of two stages $\mathcal{T} \rightarrow \mathcal{P}lan$ and $\mathcal{P}lan \rightarrow \mathcal{T}ext$.

Since plans are not observed, we need to decide how to model them. One can distinguish two strategies for this decision: (1) latent transition-based model and (2) representation-based. In the latent approach, we apply a neural encoder-decoder architecture with a transition-based model for planning similar to (Nivre et al., 2004). The neural encoder creates a latent representation of the input $\mathcal{T}$. Conditioned on this representation, the decoder works in a gradual manner to perform predefined actions that correspond to micro-planning decisions. These actions include generating a word, deciding to aggregate two relations, closing a sentence and starting to generate a new one. In this approach, there is no explicit representation of plans, instead, the model maps a latent representation of the input structure to discrete micro-planning decisions. The exact list of these decisions corresponds to the claims of a text planning theory.

In contrast, a representation-based approach separates this procedure into two well defined substeps. In a first step, the input data is mapped to a plan. This plan describes the order between atomic units into sentences, and packaging of the data into a sentence-level micro-plan. In a second step, given this plan natural text is generated. We refer to these two models as planner and realizer.

In the second approach, we must select a formalism to represent plans, which depends both on the input data and on the nature of the desired generated text. Furthermore, this approach requires data that explicitly represents such plans or from which plans can be derived to allow learning in a supervised manner. An obvious advantage of this approach is that both tasks are of lower complexity than end to end data to text which should make them easier to learn. Another advantage is better interpretability. One can measure the efficiency of plans generation and measure the correlation between quality of plans and the success of the overall task.

In this work, we explore the planner-realizer approach with a definition of plans as derived from the DeepNLG dataset. We formalize the task of WebNLG planning as follows: Given a set of triplets $\mathcal{T} = \{t_1, ..., t_m\}$ output an ordered list of ordered lists $p = (s_1, ..., s_n)$ such that $\bigcup_{i=1}^{n} s_i = \mathcal{T}$ and for each $i, j \in \{1, ..., n\}$, such that $i \neq j$, $s_i \cap s_j = \phi$.
The task consists of: (1) grouping triplets into sentences; (2) determining the order of sentences; (3) determining the order of triplets within each sentence. This definition does not provide an explicit measure of plan quality, we start by investigating what would be a good metric to assess plan quality. The choice of this simple plan formalism is an initial operational step, which has the benefit of relying on existing data. In the future, we will explore different representation formalisms for plans, and their connection to the decisions made by the realizer. We expect that document plan theories explaining information flow and packaging will provide fertile ground for this work (Kuppevelt, 1996; Roberts, 2012).

4 Plan Quality Measure

In order to maximize the benefit from the separation of planning from realization, we need to measure the intermediate success in the generation of plans. Given a tool to measure the quality of constructed plans that is known to be correlated with the quality of the output text, one can com-

pare different approaches to planning, and enhance results of a complete data-to-text pipeline.

We propose such a metric that evaluates a candidate plan against a set of reference plans as observed in the data (such as DeepNLG). Our metric combines two aspects: ordering consistency and grouping consistency. Given a plan $p = (s_1, ..., s_n)$ constructed from m triplets, denote $p_{flat} := s_1 \circ s_2, ..., \circ s_n = (t_{i_1}, ...t_{i_m})$, the ordered concatenation of all items in p. p_{flat} corresponds to the ordered list of triplets as would appear in the generated text according to plan p without considering grouping them into sentences. Given a candidate and a reference plan $\hat{p} = (s'_1, ..., s'_n)$ and $p = (s_1, ..., s_n)$ involving m triplets, we denote x_i and y_i as the positions or indices of t_i within the ordered lists $\hat{p}_{flat}$ and p_{flat} respectively for any $1 \leq i \leq m$ (both plans are of the same length – the shorter plan padded with empty lists when needed). We use Kendall's ranking correlation coefficient to define:

$$\tau(\hat{p}, p) = \frac{\sum_{i<j} sign(x_i - x_j) sign(y_i - y_j)}{\binom{m}{2}}.$$

Next we define grouping accuracy:

$$\alpha(\hat{p}, p) = \frac{\sum_{i=1}^{n} s'_i \cap s_i}{m}$$

A higher value of $\tau(\hat{p}, p)$ indicates similarity in the triplets' order of appearance between the candidate and reference plan. A higher value of $\alpha(\hat{p}, p)$ indicates similar grouping of triplets into sentences and similar ordering of sentences.

We combine these two aspects in a convex combination to define the plan quality metric PQM:

$$PQM(\hat{p}, p) = \lambda \cdot \frac{\tau(\hat{p}, p) + 1}{2} + (1 - \lambda) \cdot \alpha(\hat{p}, p)$$

When a candidate plan $\hat{p}$ is measured against a set of reference plans $\mathcal{P} = \{p_1, .., p_k\}$, we define $PQM(\hat{p}, \mathcal{P}) = \max_{1 \leq i \leq k} PQM(\hat{p}, p_i)$.

The value of the parameter λ is empirically chosen to be 0.7. In order to determine this value, we select the value which provides the highest correlation with the end-to-end text quality as measured by the BLEU metric for a realizer that takes a plan as input. To perform these steps, we train two distinct models: (1) a planner trained on the DeepNLG development set entries; (2) a realizer which generates the observed text given a DeepNLG plan as input. We train both models with a T5 text-to-text transformer model similar to (Kale, 2020). Given this pipeline, we computed PQM_λ for λ in $(0.1 \ldots 0.9)$ and the BLEU score per entry. This procedure provides PQM_λ estimations and a single BLEU score per entry in the development set. Pearson's correlation measure for each value of λ between PQM_λ and BLEU scores lets us pick the optimal λ.

model	BLEU	PQM
T5	44.44	–
T5 teacher exposure	47.50	–
T5 planner-realizer	55.01	0.838

Table 1: Model evaluation

	E2E	P+R	refs
#words	38.77	21.67	22.60
#sentences	3.42	1.34	1.45
#coordinated NPs	0.15	0.33	0.28
#coordinated VPs	0.12	0.32	0.21
#relative clauses	0.11	0.29	0.24
#subordinate clauses	0.06	0.16	0.19
#coordinate clauses	0.05	0.12	0.11

Table 2: Syntactic Phenomena Frequencies: E2E - T5 end-to-end, P+R - T5 planner + realizer

5 Experiments

We compare baseline data-to-text models which are trained to map end-to-end WebNLG 2020 data to text using the same T5 transformer-based architecture with the modular architecture (Planner, Realizer) where each of the modules is trained separately. We compare two end-to-end baselines: The first is a pre-trained T5 model which has shown promising results on data-to-text (Kale, 2020). It is fine-tuned to generate text given input triplets. In the second baseline, we use the same T5 backbone with a teacher exposure strategy during fine-tuning: each entry is composed of the input triplets as before concatenated with an incomplete prefix of the desired reference text that contains complete sentences. In this approach, the model learns to complete text given all triplets and a text prefix. The third model is the modular (Planner, Realizer) pipeline described above. Results (Table 1) indicate overall improvement in BLEU scores when using the modular approach.

To assess the impact of better controlling planning, we specifically investigate syntactic aspects of the generated text related to text cohesion (Halliday and Hasan, 1976). Indeed, planning does not determine all aspects of realization - for example, it does not impact the lexicalization of entities encoded in triplets, but it does impact directly sen-

tence packaging, aggregation, coordination, clause structure, relative clauses. Table 2 compares text generated by the baseline end-to-end model, the modular model (Planner, Realizer) and the reference texts of WebNLG 2020. It shows the frequency of different syntactic phenomena per sentence in the text as well as number of words and sentences in each text. To identify occurrences of these phenomena, we used spaCy's (Honnibal and Montani, 2017) dependency parser along with rule-based methods to identify each configuration.

We observe that the end-to-end baseline model generates much longer text (both number of words and sentences). Manual inspection shows that it introduces repetitions that are avoided in the planner-realizer approach. Another notable finding is that the frequencies of cohesive devices in texts generated by the planner-realizer model are much more similar to those in the reference texts than the end-to-end approach.

In this study, we illustrated a computational approach to justifying a modular approach for data to text generation. Starting with a very simple representation model of plans (as a sequence of groups of triplets), we specified a learnable text plan module and an evaluation metrics for the generated plans. We demonstrated empirically that a modular model separating planning and realization generates more cohesive text with less repetitions. We believe this empirical platform opens routes for fruitful exchange with more sophisticated text planning models and their interaction with realization.

References

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Albert Gatt and E. Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *JAIR*, 61.

M. A. K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks.

Jan Van Kuppevelt. 1996. Directionality in Discourse: Prominence Differences in Subordination Relations. *Journal of Semantics*, 13(4):363–395.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Craige Roberts. 2012. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics*, 5(6):1–69.

Annotating QUDs for generating pragmatically rich texts

Christoph Hesse, Anton Benz
Leibniz-Zentrum Allgemeine Sprachwissenschaft
{last name}@zas-berlin.de
Maurice Langner, Felix Theodor, Ralf Klabunde
Ruhr-Universität Bochum, Department of Linguistics
{first name.last name}@rub.de

Abstract

We describe our work on QUD-oriented annotation of driving reports for the generation of corresponding texts – texts that are a mix of technical details of the new vehicle that has been put on the market together with the impressions of the test driver on driving characteristics. Generating these texts pose a challenge since they express non-at-issue and expressive content that cannot be retrieved from a database. Instead these subjective meanings must be justified by comparisons with attributes of other vehicles. We describe our current annotation task for the extraction of the relevant information for generating these driving reports.

1 Introduction

Driving reports about new vehicles, typically published in national daily newspapers and online journals, constitute a text type that poses a challenge for NLG systems since these texts express technical details about these vehicles (often in comparison with previous models or alternative vehicles) combined with subjective impressions of the test driver, resulting in a number of expressive and evaluative expressions. To illustrate these phenomena, we show the English translation of an excerpt from a German driving report about the Porsche Cayenne Turbo S E-Hybrid:

1. *With the Turbo S E-Hybrid strand, Porsche has made a very clever move. The top model, of all models, is no longer the greedy bogeyman, but is ecologically sound when used appropriately. One can of course smile at the statement of the combined consumption of 3.7 litres per 100 km and revile the basis for calculation, but provided that the four-wheel drive car is driven electrically, this value can also be achieved in real terms. Whether this is environmentally friendly or not, especially since electricity is by no means only generated from sun or wind, is another matter.*
   (Original text: Mit dem Turbo S E-Hybrid-Strang hat Porsche einen durchaus cleveren Schachzug gemacht. So ist ausgerechnet das Topmodell nicht mehr der gefräßige Buhmann, sondern schlägt sich bei entsprechender Nutzung ökologisch wacker. Man kann die Angabe des kombinierten Verbrauchs von 3,7 Litern je 100 km natürlich belächeln und die Berechnungsgrundlage schmähen, aber unter der Voraussetzung, den Allradler fleißig elektrisch zu fahren, kann dieser Wert auch real zustande kommen. Ob das umweltfreundlich ist oder nicht, zumal Strom bekanntermaßen keineswegs nur aus Sonne oder Wind gewonnen wird, steht auf einem anderen Blatt.)

This text contains facts about consumption and drive type, but most of the text is about subjective estimations and appraisals, realized by evaluative adjectives (*greedy*), adverbs (*of course, by no means*), the use of metaphors (*bogeyman*), and other expressive-related linguistic means. Our research question concerns the relationship between facts and evaluations in driving reports and the justified use of subjective, expressive content in generating these reports.

In generating driving reports, we aim at the explanatory power of *Question under Discussion* (QUD) accounts to text structuring and textual development (van Kuppevelt, 1995; Roberts, 1996). QUD approaches assume that texts are answers to a structured set of explicit or implicit questions. Each QUD does not only impose constraints on the propositional content of a single sentence, but it also determines the focus/background structures and the distinction between at-issue (material that helps to answer the QUD) and non-at-issue content (everything else, typically including evaluative and expressive content). By this, QUD-based approaches provide strong hypotheses about the textual development by successively answering the corresponding QUDs.

These theories provide the starting point of our work: Based on the theory-driven assumptions on textual development, we are going to generate driving reports that are as close to the original texts as possible. If specific content cannot be systematically determined based on QUD requirements, we

get evidence for shortcomings of the underlying theory. Hence, we intend a kind of 'reverse generating' to test the adequacy of QUD-based theories.

The fundamental source for this approach are QUD-annotated data where the QUDs reflect the informational needs to be satisfied by section of the text down to single sentences. It is well known that annotating QUDs in texts requires an intensive training of the annotators and sophisticated annotation guidelines in order to receive reliable results (Arndt Riester and Kuthy, 2018), but if these QUDs have been formulated properly, they provide strict information-structural constraints for their answer that can be used in the generation process. Therefore, we will introduce the underlying data, the problems that occur in annotating the driving reports, and first results concerning resulting QUD structures.

2 Underlying data

We selected 40 driving reports from German online journals (faz.net; welt.de).

The 40 vehicle reports in the corpus were annotated for QUDs and sub-QUDs, focus/background, and non-at-issue content. The guiding principle for us was to be able to separate purely propositional content from non-at-issue, evaluative and expressive content (following Roberts, 1996; Büring, 2003).

QUDs are well-accepted as a discourse-structuring device (Carlson, 1983; van Kuppevelt, 1995; von Stutterheim, 1997). More importantly for us, however, QUDs have been identified as a crucial criterion for distinguishing between at-issue and non-at-issue content. Content which may be non-at-issue with respect to its immediate sub-QUD may nevertheless supply relevant information in the context of an over-arching super-QUD. Analyzing the depth of embedding may give useful insight to enable us to anticipate follow-up QUDs to incomplete sub-QUDs (Onea, 2016).

Crucially for us, QUD approaches recognize that discourse is not merely a coherent presentation of relevant information, but its structure is goal-oriented. Authors consciously decide which questions they want to address, how they want to address them and in what order. Similarly authors make conscious choices about the use of rhetorical relations such as elaboration, contrast, and concession as text-structuring devices, which may or may not be fully predictable from QUD-trees. RST-trees reveal recurring rhetorical structures with predictable evaluative and expressive effect (e.g., employing a contrast relation whenever a vehicle's shortcomings are compensated by other positive attributes, or when comparing a vehicle to its competitors).

We also use QUDs to annotate focus (cf. Roberts, 1996; van Kuppevelt, 1995; von Stutterheim, 1997). Büring (2003) applied Robert's (1996) QUD-stack model to contrastive focus using a QUD-tree model, which under certain conditions can lead to tree structures similar to RST-trees. We use the QUD-trees to arrive at a complete representation of discourse structure. Sub-QUDs are divided further into sub-QUDs until each terminal sub-QUD can be directly mapped to a database entry (e.g., *What is the vehicle's rate of acceleration?* for numerical values or *What type of transmission does it use?* for referring expressions).

The QUD at the root of the tree of each text answers the question whether the vehicle is qualitatively good and worth purchasing. From this root QUD, immediately a sub-QUD derives: Compared to other vehicles or a standard, where the search space is dynamically populated with comparison objects given explicit references in the text (other vehicles in the same class or from competitors) and implicit comparisons based on attribute scales (e.g., vehicles with a comparable type of use or type of engine, transmission, interior, drive assistance features, etc.). We also assume that the typical sections of vehicle reports provide partial answers to the root QUD: a section about the engine, acceleration, gas consumption, mileage, etc. speaks to the quality of the vehicle's performance; a section about the interior speaks to the level of comfort; the test drive speaks to how well the performance promised by the manufacturer holds up under real-world driving conditions, and so forth.

Since an author's subjective view of a vehicle has a tremendous impact on text and information structure, and specifically on the foregrounding and backgrounding of certain information available about the vehicle in the database, capturing authors' subjective evaluation can lead to vastly different QUD phrasings and QUD-structures across annotators. Thus, inter-annotator agreement is the biggest challenge in arriving at systematic discourse structures within this text genre.

The concrete annotation work points to some fundamental problems concerning the assignment

of corresponding tags. Some of them refer to shortcomings of QUD-oriented theories, but others are of a language-specific nature. The main problems that occurred during the annotation process concern so-called feeders (van Kuppevelt, 1995), implicit correlations for contrastive relation, and the assignment of focus.

2.1 Focus

Focus and its complement, background, are information-structural notions that express that part of an utterance that is new to the hearer vs. the information already known by her. In the context of QUD-based theories we could identify that part of a sentence that answers a QUD as focused while the rest belongs to the background.

In many languages focus is expressed phonologically by a lexical item carrying a focus-related accent, the focus exponent. Focus theories explain the position of the focus-related accent by the percolation of a focus feature [F] in a syntactic tree to the bearer of the focus accent. The constituent that expresses the focus, however, is often more extensive that the focus exponent which gives rise to focus ambiguity, i.e. the problem to determine that constituent, given the focus exponent, that expresses the focused information.

For example, the sentence *Anne likes to test the PORSCHE* with *Porsche* being the focus exponent, we have at least two possible focus constituents:

(1) *Anne [likes to test the PORSCHE]$_F$*

(2) *Anne likes to test [the PORSCHE]$_F$*

Which one of these constituents expresses the focus depends on the QUD that shall be answered. Sentence (1) answers the QUD *What's new about Anne?* while the second sentence answers the question *What does Anne like to test?*

In addition to accent placement for expressing focused information, languages provide focus particles and specific syntactic constructions for expressing which part expresses the focus. German (and English) provides all three possibilities, but accent placement is the most prominent means for signaling focus.

The relation between focus and the linguistic means for expressing it seems to be transparent so that annotating focus, once the QUD has been established, should not be a major effort. However, our data indicate some problems in focus assignment that have direct repercussions for the development of the annotation tool and some focus theories as well.

Split-focus: In our data, some sentences have two focus constituents that express one focus together. For example, one QUD in a driving report is *What about the power unit?* The sentence that answers this QUD in the text is:

(3) *In der praktischen Außenhaut des 3,60 kurzen Fünftürers war der Antrieb erstmal kaum zu erkennen.*
'In the practical outer skin of the of the 3.60 short five-door car, the power unit was hardly noticeable at first.'

A plausible assignment of focus is to tag *In der praktischen Außenhaut* ('In the practical outer skin') and *war der Antrieb erstmal kaum zu erkennen* ('the power unit was hardly noticeable at first') as being focused, but not 'the 3.60 short five-door car' since this constituent doesn't provide a part for the answer to the QUD. The consequence of these finding, which have hardly been mentioned in the literature, was to adjust the annotation tool to these phenomena by introducing the possibility to set indexes by the annotators in order to express that both foci belong together.

A further but related phenomenon concerns sentences consisting of two coordinated main clauses, each with its own focus, but answering one QUD:

(4) QUD: *How is the Renault Captur?*
Der Renault Captur [wächst]$_F$ und [verändert seinen Charakter]$_F$.
'The Renault Captur grows and changes its character.'

Is is reasonable to assume two separate foci since this coordination refers to two new aspects of the tested car. Both foci are well motivated by the QUD; they demonstrate that one QUD does not necessarily set up one focus only. Ellipses also indicate that the "one QUD – one focus" default can be violated:

(5) QUD: *How have the aesthetics changed, compared to the old Captur?*
[das sieht scharf trainiert]$_F$ und [angriffslustig aus]$_F$.
'that looks sharply trained and ready to attack.'

The non-elliptic sentence in German would be *das sieht scharf trainiert aus und das sieht angriffslustig aus*, with the prefix *aus* separated from the

prefix verb *aussehen* and remaining in the base position, and the subject plus verb stem inserted in the second clause. The ellipsis forces an index as well for expressing that both foci belong together; otherwise the ellipsis cannot be handled correctly.

The final example illustrates the complexity of focus and background tagging with respect to information-structural considerations. In this example one sentence answers two, actually unrelated, QUDs:

(6) QUD: *How is the interior?*
Der Innenraum macht [einen Sprung in eine neue Zeit]$_F$,
'The interior takes a leap into a new era,'

> QUD: *What will make the new era much more pleasant?*
> *die [mit digitalen Instrumenten, großem Bildschirm, schwebender Mittelkonsole, feineren Oberflächen und adretteren Schaltern]$_F$ deutlich angenehmer wird.*
> 'which becomes much more pleasant with digital instruments, large screen, floating center console, finer surfaces and neater switches'

In (6) one sentence from the driving report answers two QUDs formulated by the annotator. The first one will be answered by the focused constituent in the main clause. In order to motivate the relative clause, a new QUD has been stated and the list of attributes of the car functions as focus.

2.2 Feeder sentences

Another challenging aspect the annotation illustrates is the fact that not every sentence answers a QUD at all. An example for this would be a segment like *Nun war ein größerer Schritt fällig* ('It was time for the next big step'), which shifts the topic of the text in a certain direction but does not provide any information relevant to a QUD. Van Kuppevelt (1995) defines this as "a topicless unit of discourse, [...] or one whose topic is no longer prominent at the moment of questioning". We follow his definition and call these segments (linguistic) feeders. Feeders constitue a trigger for QUDs to arise, but they are not motivated by QUDs themselves. Their status seems to be outside the scope of QUD-based theories.

The example below demonstrates that. Since the given context does not require any information about sale figures of former cars, the segment cannot be motivated by a QUD. However, this new information leads to other QUDs arising, because it provides a set of indeterminacies to which there is no information in the given context:

Feeder: *1,2 Millionen Captur sind seit 2013 verkauft worden.*
'1.2 million Captur have been sold since 2013.'

> QUD: *What about the first generation of Captur?*
> QUD: *What did it look like?*
> *Am Anfang mit trüben Scheinwerfern und viel hartem Kunststoff [...]*
> 'With cloudy headlights and a lot of hard plastic in the beginning [...]'

Feeder sentences often function as an introduction to a new topic, therefore most of them can be found at the start of a new paragraph or unit of text. As van Kuppevelt (1995) notes, even segements that provide information relevant to a QUD can technically act as a feeder as well (if they raise new questions), but we restrict the annotation of feeders to cases in which their appearance is clearly not motivated by a QUD.

2.3 Contrast

QUD approaches emphasize the goal-oriented nature of a text's information structure. Authors' primary goal in the driving report genre is to evaluate a vehicle based on its overall qualities. In order to arrive at an overall evaluation, authors examine inidivual topic areas such as technical specifications, driving experience, comfort, and accessories in turn. Often times authors will note that outstanding performance in one area compensates for deficits in other areas, or that performance in one area is striking compared to previous models or competitors. This makes *Contrast* one of the most common discourse relations found in driving reports, and authors use a variety of surface realizations to express contrast without marking it overtly (no use of the constrastive marker *but*). The following example shows some of these strategies:

1. *Harmonious gliding or hard driving at the limit, the GS, which has become five kilograms heavier, masters both without any efforts. Fortunately, the BMW developers succeeded not only in improving the quality of the exhaust gases, but also in reducing fuel consumption by 0.2 litres/100 km: Despite the fact that the driving style was by no means restrained, the lavishly equipped on-board computer of the test bike showed just 4.8 litres*

per 100 kilometres.
(Original text: Harmonisches Gleiten oder hartes Fahren am Limit, beides beherrscht die um fünf Kilogramm schwerer gewordene GS quasi mit links (π_1). Erfreulicherweise gelang es den BMW-Entwicklern zugleich, nicht nur die Abgasqualitaät zu verbessern, sondern auch den Verbrauch um 0,2 Liter/100 km zu reduzieren (π_2): Trotz keineswegs zurückhaltender Fahrweise zeigte der üppig bestückte Bordcomputer des Testbikes gerade mal 4,8 Liter pro 100 Kilometer an (π_3).)

The evaluative adverb and discourse marker *erstaunlicherweise* (fortunately) marks a *Contrast* relation, but note this relation does not hold between two explicit propositions in the text, rather it holds between (i) the conjoined explicit propositions π_2 and π_3, and (ii) the *unforefilled* implicit expectation of higher gas consumption (and with that poorer exhaust quality), expectations raised by the appositive *um fuenf Kiligramm schwerer gewordene* (weight increase of 5 kg) in π_1. (Simons et al., 2011) claim that appositives are not-at-issue because they do not speak to the QUD answered by the matrix clause which contains the appositive. However, the appositive *um fünf Kilogramm schwerer gewordene* is only *locally* not-at-issue because globally it is very much at-issue for the Contrast relation that follows.

The use of the evaluative adverb and contrastive marker *erstaunlicherweise* (fortunately) is licensed by positive surprisal. Surprisal presupposes a difference between the expected and the actual, and when this difference is positive, i.e. when the actual surpasses the expected, the surprisal is positive and the adverb is licensed. If the new model of the BMW bike consumes 0.2 L/100 km less than the previous model, the previous model consumed 5 L/100 km. Because of the new model's higher weight, its expected gas consumption should be >5 L/100 km. So the surprisal is two-fold: (1) the actual consumption is less compared to the previous model ($4.8 < 5$), and (2) it is less compared to the consumption expected due to the bike being heavier than the previous model ($4.8 < [> 5]$). Since the new model consumes both less than the previous model and less than expected due to weight, *erstaunlicherweise* (fortunately) is double-lincensed. Mentioning the hard driving conditions during testing only emphasizes the level of surprise, while the explicit mention of the onboard computer emphasizes the reliability of the measurements.

Crucially, It is the appositive in π_1 which raises (or at least explicitly adds to) this expectation of higher gas consumption (hard driving conditions $\rightarrow$ higher consumption $\wedge$ higher weight $\rightarrow$ higher consumption). The joint-marker *nicht nur ... sondern auch* (not only ... but also) introduces the two consequents in π_2: gas consumption and exhaust gas quality. The explicitly mentioned weight increase raises causal expectations: higher weight $\rightarrow$ more gas consumption $\rightarrow$ more exhaust gases $\rightarrow$ poorer exhaust gas quality. The contrast relation holds for both the surprisingly good exhaust gas quality and the bike's gas consumption. Both of these implicit contrasts require the assumptions raised by the appositive in π_1. So while the appositive locally may be not-at-issue for how the bike handles, it must be globally at-issue to explain the overtly marked *Contrast* between expected higher gas consumption (and poorer exhaust quality) and the surprisal of actual gas consumption (and exhaust) being lower.

The implicit *Contrast* suggests that the topical QUD of this text should be something like 'Why was the reduction of gasoline consumption surprising/unexpected?' But this would mean the *embedded* appositive needs to be structurally on an equal level with the reduction propositions π_2 and π_3 while the QUD of the matrix proposition in π_1 should be something like 'How does the bike handle under smooth and hard driving conditions?' So while embedding the appositive suggests that the matrix clause's QUD supersedes the appositive's relevant QUD, the *Contrast* relation makes it clear that the QUD hierarchy is actually inverse to the embedding structure. We find this sort of *Contrast* relation with implicit expectations and causal relations raised by technical details quite frequently in our corpus. Our hope is that proper QUD structures which capture the implicit expectations can enrich debates about contrast marking (Jasinskaja and Zeevat, 2008) and information structure (Umbach, 2005).

3 Text planning

Our preliminary analysis of the corpus shows that, broadly speaking, vehicle reports are divided into three parts: (1) an introduction, which may give background information on the manufacturer, occasion for the new release (e.g., anniversaries), stylistic or technical choices characteristic of the vehicle situated in a line of previous models or the history of the line; (2) a main part, which consists of (2a) general technical specifications as advertized by the manufacturer and (2b) impressions from the test drive; (3) an outro which may include price

listings for different models of the vehicle (plus accessories), release dates or additions/changes the manufacturer is planning before the release. Part (2a) usually tends to focus on the most crucial technical details, especially changes which have been made compared to previous models. Part (2b), in stark contrast, is usually a visceral, metaphor- and idiom-rich description of the driving experience aimed at emotionally immersing the reader.

The more engaging these texts are, the more they deviate from this generic structure: Aspects of the vehicles which are exceptionally good or exceptionally bad are foregrounded. We will predict striking features of vehicles via pair-wise feature comparison to other vehicles in the same category. Given a large comparison class, 'average' features will cluster normally around a mean along an evaluation dimension (e.g., less gas consumption is better) while expectable features will correspond to extreme values on either tail of the average distribution. An exceptionally good vehicle excels in all categories that the generic structure explicitly discusses. When a vehicle does not tick all its boxes, authors often restructure the text to make clear how certain excellent features in some categories make up for the shortcomings in other categories. Authors also make a conscious decision to note positive things about a vehicle before diving into its shortcomings, and they try to end on a positive note.

Evaluating the technical details in our database along quality dimensions by comparing vehicles against other vehicles *as well as* comparing different aspects of a vehicle with other aspects of the same vehicle is fundamental for our approach. The overall evaluation made in a vehicle report is the sum of the evaluation of its individual aspects. Not all technical details contained in the ADAC database are explicitly mentioned in the reports, and of those that are mentioned, some are given more weight than others in contributing to the overall evaluation. We aim to derive this weighting probabilistically from the comparison analysis of technical features in the database. Since quality dimensions associated with these features are subjective, these are based on the original annotation.

The type of vehicle (e.g., ICE, internal combustion engine, versus EV, electric vehicle; car versus motorbike) pre-selects a subset of relevant technical features as well as a class-specific document plan. We then go through the evaluation process as described above. The result of this process is a linearized text plan with vehicle features weighted for relevance and impact on the overall evalutation. We assume that non-at-issue content does not directly answer a proposition's immediate QUD, but, instead, it contextualizes the choices authors make in establishing the foregrounding and backgrounding of vehicle features and marks subjective evaluation. With the evaluation process complete, the text plan can be enriched with non-at-issue content.

4 Surface realisation

For realizing the sentences, a hierarchy of classes has been set up which defines messages for categorical pieces of information that are stereotypically produced in the genre of vehicle reports, e.g. 'HorsePowerMSG'. Each of these classes may perform its own lexicalisation task by a proper interface function. A microplanner class provides containers for messages, on which aggregation tasks and other post-processing may operate.

Among those post-processors, a module for referring expression generation and coreference realization are going to be implemented. Across the microplan, references to the object under discussion are filled with placeholders. A suitable method for this is based on the QUD structure and the depth of embedding of paragraphs, which limit the availability of entities and prevents the usage of pronominal reference. A focus-stack model keeps track of mentioned entities and the different lexicalisation options for the object at the given position.

A lexicon is built from the corpus including idioms, which allows for a probabilistic distribution over head verbs that may be used to lexicalize different messages. The subcategorization frames allow to further process both syntactic and morphologic processes.

Simplifications must be made according to background knowledge and authors' opinions regarding different cars, which would demand for a complex common sense reasoning database. Instead, we intend to use predefined templates for these portions of text in order to achieve our aim of showing whether this non-at-issue content can be generated.

For surface realisation, we use the Java library of SimpleNLG for German (Bollmann, 2011), which covers mainly morphological operations. An interface between micro-planning and SimpleNLG is needed in order to call the correct methods for the respective syntactic constructions defined by the

micro-planner. This means that an interpreter for the micro-planning implements both a linearization of lexemes and a mapping from AVM structure to Java methods in SimpleNLG.

5 Summary and outlook

Without annotations with sufficient quality, one cannot generate good texts. We are interested in adopting the QUD-approach to text structuring to generating reports in order to test the soundness of this approach. QUD-based linguistic analyses tend to be confined to simplified texts with a focus on relevant phenomena; we want to know whether such a theory-driven approach to generating pragmatically rich texts is feasible.

References

Liesa Brunetti Arndt Riester and Kodula De Kuthy. 2018. *Annotation guidelines for Questions under Discussion and information structure*, pages 403–443. Benjamins, Amsterdam.

Marcel Bollmann. 2011. Adapting SimpleNLG to German. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG 2011)*, page 133–138, Nancy.

Daniel Büring. 2003. On D-trees, beans, and B-accents. *Linguistics & Philosophy*, 26(5):511–545.

Lauri Carlson. 1983. *Dialogue Games: An Approach to Discourse Analysis*. Reidel, Dordrecht.

Katja Jasinskaja and Hank Zeevat. 2008. Explaining additive, adversative and contrast marking in russian and english. *Revue de Sémantique et Pragmatique*, 24:65–91.

Jan van Kuppevelt. 1995. Discourse structure, topicality and questioning. *Journal of Linguistics*, 31:109–147.

Edgar Onea. 2016. *Potential Questions at the Semantics-Pragmatics Interface*, volume 33 of *Current Research in the Semantics/Pragmatics Interface*. Brill, Leiden.

Craige Roberts. 1996. Information structure in discourse: Toward an integrated formal theory of pragmatics. In Jar Hak Yoon and Andreas Kathol, editors, *OSU Working Papers in Linguistics*, volume 49, pages 91–136. The Ohio State University, Department of Linguistics, Ohio.

Mandy Simons, Judith Tonhauser, David Beaver, and Craige Roberts. 2011. What projects and why? *Semantics and Linguistic Theory*, 20:309–327.

Christiane von Stutterheim. 1997. *Einige Prinzipien des Textaufbaus: Empirische Untersuchungen zur Produktion mündlicher Texte*, volume 184 of *Reihe Germanistische Linguistik*. Niemeyer Verlag, Tübingen.

Carla Umbach. 2005. Contrast and information structure: A focus-based analysis of *but*. *Journal of Linguistics*, 43:207–232.

Towards Domain-Independent Text Structuring Trainable on Large Discourse Treebanks

Grigorii Guz and Giuseppe Carenini
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{gguz, carenini}@cs.ubc.ca

Abstract

Text structuring is a fundamental step in NLG, especially when generating multi-sentential text. With the goal of fostering more general and data-driven approaches to text structuring, we propose the new and domain-independent NLG task of structuring and ordering a (possibly large) set of EDUs. We then present a solution for this task that combines neural dependency tree induction with pointer networks and can be trained on large discourse treebanks that have only recently become available. Further, we propose a new evaluation metric that is arguably more suitable for our new task compared to existing content ordering metrics. Finally, we empirically show that our approach outperforms competitive alternatives on the proposed measure and is equivalent in performance with respect to previously established measures.

1 Introduction

Natural Language Generation (NLG) plays a fundamental role in data-to-text tasks like automatically producing soccer, weather and financial reports (Chen and Mooney, 2008; Plachouras et al., 2016; Balakrishnan et al., 2019), as well as in text-to-text generation tasks like summarization (Nenkova and McKeown, 2012).

Generally speaking, NLG involves three key steps (Gatt and Krahmer, 2017): first there is content determination which selects what information units should be conveyed, secondly there is text structuring, which is responsible for properly structuring and ordering those units; and finally microplanning-realization that aggregates information units into sentences and paragraphs that are then verbalized.

The focus of this paper is on the text structuring step, which is critical for the overall performance of an NLG system, as it ensures that the communicative goals of the text are realized in the most structurally coherent and cohesive way possible, making the main ideas expressed by the text easy to follow for the target audience.

Aiming to develop very general computational methods for text structuring, we keep our study independent from particular ways in which the input information units are represented and from explicitly provided ordering constraints for the target application domain (Gatt and Krahmer, 2017). More specifically, we propose and attack, in a fully data-driven way, the novel and domain-independent task of simultaneously structuring and ordering a set of Elementary Discourse Units (EDUs), i.e., clause-like text fragments that the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) assumes to be the building blocks of any discourse structure (see Figure 1(a)(left)). In other words, we assume that the system is given a set of EDUs (with cardinality possibly > 100) as input and returns their ordering, as well as the unlabelled RST dependency discourse tree structure for a document consisting of this set of EDUs, as illustrated in Figure 1(a).

Our data-driven approach relies on the very recent availability of large treebanks containing hundreds of thousands of (silver-standard) discourse trees that can be automatically generated by distant supervision following the approach presented by Huber and Carenini (2020). We formulate the problem as one of the dependency tree induction, repurposing existing solutions (Ma and Hovy, 2017; Vinyals et al., 2015) to perform an RST-based text structuring where both EDU ordering and tree building are executed simultaneously (Reiter and Dale, 2000). The resulting structures can be highly useful for subsequent NLG pipeline stages such as aggregation, and for downstream tasks like text simplification (Zhong et al., 2019). Our approach is trainable end-to-end, but since the discourse trees in the training treebank are constituency trees (see

Figure 1(b)), we face the additional challenge of turning them into dependency trees (see Figure 1(a)) before the learning process can start (Hayashi et al., 2016).

In a comprehensive evaluation, we compare our solution to three baselines along with a competitive approach based on pointer networks (Vinyals et al., 2015), which is the established method of choice not only for sentence ordering (Cui et al., 2018), but also for basic domain-specific text structuring in data-to-text applications (Puduppully et al., 2019). In particular, the comparison involves training and testing the different models on the MEGA-DT treebank (Huber and Carenini, 2020), containing ≈250,000 discourse trees obtained by distant supervision from a the Yelp'13 corpus of customer reviews (Tang et al., 2015).

With respect to evaluation metrics, we found the current ways of measuring content ordering (e.g., Kendall's τ) to be inadequate to capture the quality of long sequences of relatively short information units (i.e., sequences of EDUs of long multi-sentential text). Thus, we propose a novel evaluation measure, Blocked Kendall's τ, that we argue should be used for our new NLG task of ordering and structuring a possibly large set of EDUs, because it critically measures how well semantically close units are clustered together in the correct order.

To summarize the contributions of this paper: **(i)** we propose the new and domain-independent NLG task involving the structuring and ordering a set of EDUs, which is intended to enable more general and data-driven approaches to text structuring; **(ii)** we present a strong benchmark solution for this task, trainable on large discourse treebank, that combines neural dependency tree induction with pointer networks; **(iii)** we propose a new evaluation metric that is arguably much more suitable for this task than existing ordering metrics; **(iv)** and on this new metric along with standard tree-quality metrics, we show empirically that our approach outperforms or is comparable to competitive alternatives. The code for our solution and the new metric, as well as the treebank for training, is publicly available.[1]

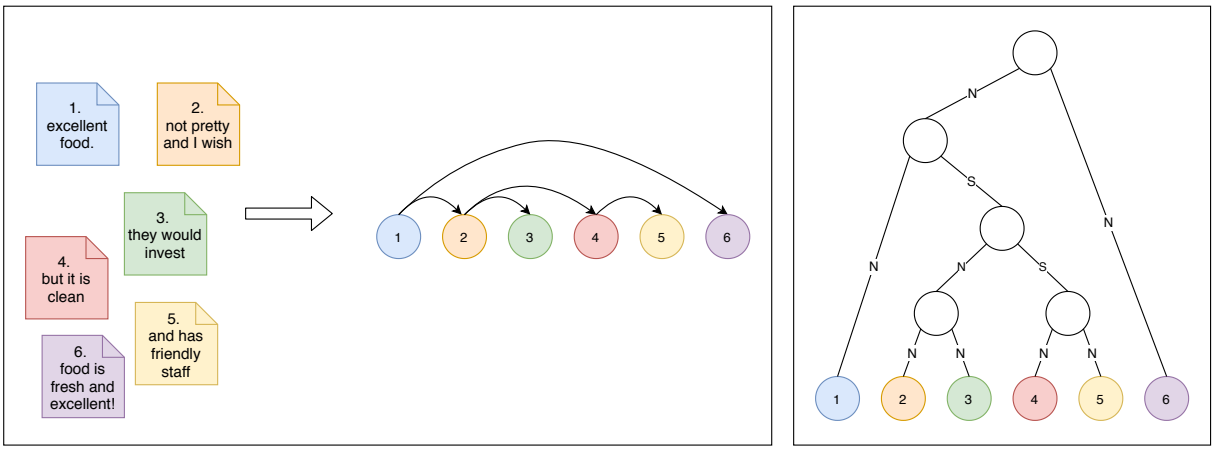
[1] http://www.cs.ubc.ca/cs-research/lci/research-groups/natural-language-processing/index.html

2 Related Work

(a) Text structuring is a key step in NLG, especially when generating long multi-sentential documents. Not surprisingly, this is also the case in recent neural approaches. Wiseman et al. (2017) presented the RotoWire corpus, targeting long-document data-to-text NLG. To generate the document, their model conditions on all records in the data table by weighting their embeddings with attention, in addition to using copying mechanism for out-of-vocabulary data entries. The follow-up work of Puduppully et al. (2019), instead of conditioning on all records, arguably performs better text structuring by first selecting and then ordering the entries of a data table using Pointer network architecture (Vinyals et al., 2015). That way, the surface realization module considers previously generated text and only one new data table entry at a time. Their model was extended by Iso et al. (2019), with an additional GRU for tracking the entities that the model already referred to in the past. Pursuing a rather different approach to improve text structuring, Shao et al. (2019) proposed a hierarchical latent-variable model where the problem is decomposed into dependent sub-tasks, aggregating groups of data table entries into sentences first and then generating the sentences sequentially, conditioned on the plan and already generated sentences. Overall, these last three models significantly outperform the initial approach of Wiseman et al. (2017) both in terms of fluency and coverage, with increasing sophistication of the text structuring module yielding bigger gains, confirming that text structuring is indeed crucial for generating coherent long documents.

The task we propose and investigate in this paper can be seen as pushing this line of research even further. We aim for a more ambitious text structuring module inspired by traditional NLG work, viewing the process as the construction of an RST discourse tree for the target document (Reiter and Dale, 2000), which critically includes assigning importance to each constituent. Tellingly, our task is also domain-independent and agnostic on the representation of the input information units.

(b) The goal of **sentence ordering** is to sort a given set of unordered sentences into a maximally coherent document. Most recent work on sentence ordering (Logeswaran et al., 2016; Cui et al., 2018; Wang and Wan, 2019) involves constructing contextualized order-agnostic representations of indi-



(a) (b)

Figure 1: (a) A simple example of the novel NLG task we propose in this paper: generating an ordered discourse dependency tree (right) for a given set of EDUs (left). (b) The constituency discourse tree corresponding to the dependency tree shown in (a). The RST-style discourse trees in the treebanks we use for our experiments are initially represented as constituency trees.

vidual sentences and full documents using architectures such as Transformer Encoder without positional embeddings (Vaswani et al., 2017), and then feeding those representations into a pointer-based decoder (Vinyals et al., 2015).

The new task we propose in this paper is similar, but more challenging than sentence ordering. Instead of ordering sentences, we need to order EDUs, which are often shorter sentence constituents, and therefore by expressing smaller semantic units they arguably require more fine-grained processing. Furthermore, our task goes beyond ordering by also requiring the synergistic and simultaneous step of generating the RST discourse structure for the EDUs. To address these challenges, more powerful techniques for tree induction are needed on top of pointer networks.

(c) Document discourse tree structure induction: The third related line of research involves the induction of latent tree structures over documents. Some of these works aim at obtaining better document representations for tasks such as text classification (Karimi and Tang, 2019) and single-document extractive summarization (Liu et al., 2019). In essence, a neural framework is designed so that a discourse tree for a document is induced while training on the target downstream task. However, even if these approaches demonstrated improvements over non-tree-based models, subsequent studies have shown that the resulting latent discourse dependency trees are often trivial and too shallow (Ferracane et al., 2019). In contrast, recent work on distant supervision from sentiment (Huber and Carenini, 2020) indicates that large treebanks of discourse trees can be generated by combining neural multiple-instance learning (Angelidis and Lapata, 2018) with a CKY-inspired algorithm (Jurafsky and Martin, 2014). Since a series of experiments in inter-domain discourse parsing have certified the high-quality of these treebanks, we use one of such treebaks, called MEGA-DT, for training and testing our data-driven text structuring approach.

3 Novel Task and Methods

Our novel task for text structuring receives as input a set of n EDUs and returns both an ordering and a discourse structure for that set. We first describe how the EDUs are encoded, as this is the initial step for all the approaches we consider. Then, after discussing a basic method for just ordering the input EDUs (Pointer Networks), which will serve as our main baseline, we present our solution for fully solving the task in detail, which combines tree induction with pointer networks. We will refer to our final approach as *DepStructurer*. We conclude the section with two simple baselines for EDU ordering and structuring, respectively.

3.1 EDU Encoder

For a clear comparison of tree vs. non-tree based approaches, we encode EDUs in a very similar way to previous sentence ordering works (Cui et al., 2018; Wang and Wan, 2019). Given a document

with n EDUs $e_{1:n}$, with each EDU e_i containing a list of m_i words $w_{1:m_i}$, the final output of the EDU encoder is a set $v_{1:n}$, $v_i \in \mathbb{R}^d$ of embeddings of its EDUs. First, using the base version of the ALBERT language model (Lan et al., 2020), we construct individual EDU embeddings $b_i \in \mathbb{R}^{768}$ as the means of EDU word embeddings $\hat{w}_{1:m_i}$ from the last layer of ALBERT:

$$b_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \hat{w}_j \quad (1)$$

This language model was chosen because it uses a sentence-ordering objective during pre-training, see Lan et al. (2020). The EDU embeddings are then fed into a Transformer Encoder (Vaswani et al., 2017) without positional embeddings, yielding contextualized EDU representations $v_{1:n}$:

$$v_{1:n} = \text{TransformerEncoder}(b_{1:n}) \quad (2)$$

As Cui et al. (2018), we compute the final document representation z by averaging the document's EDU embeddings $v_{1:n}$.

3.2 Predicting Order Only: Pointer Networks

Pointer networks are commonly used for sentence ordering tasks (Cui et al., 2018) and have been recently applied to basic text structuring in data-to-text NLG (Puduppully et al., 2019). We train a pointer network to maximize the probability of correct ordering o^s of an unordered set of EDUs $v_{1:n}$ as a sequence prediction:

$$P(o^s|v_{1:n}) = \prod_{i=1}^{n} P(o_i^s|o_{i-1}^s, ..., o_1^s, v_{1:n}) \quad (3)$$

Here, each term in the product of probabilities is computed as:

$$h_j, c_j = LSTM(h_{j-1}, c_{j-1}, v_{i-1}) \quad (4)$$

$$u_i^j = k^T tanh(W_1 v_i + W_2 h_j) \quad (5)$$

$$p(o_i|o_{i-1}, .., o_1, s) = softmax(u_i) \quad (6)$$

where $k \in \mathbb{R}^d$ and $W_1, W_2 \in \mathbb{R}^{d \times d}$ are learnable parameters and $i, j \in (1, ..., n)$ index into input and output sequences respectively. Similarly to (Vinyals et al., 2015), we use the document embedding vector z as the initial hidden state and a vector of zeros as the first input to the pointer network. More specifically, during training, for each document s in our dataset D we feed in the EDU embeddings v_i according to the gold document order o^s and maximize the probability according to

$$\theta^* = \arg\max_{\theta} \sum_{s \in D} \log p(o^*|s, \theta) \quad (7)$$

During inference, since an exhaustive search over the most likely ordering is intractable, we use beam search for finding a suboptimal solution.

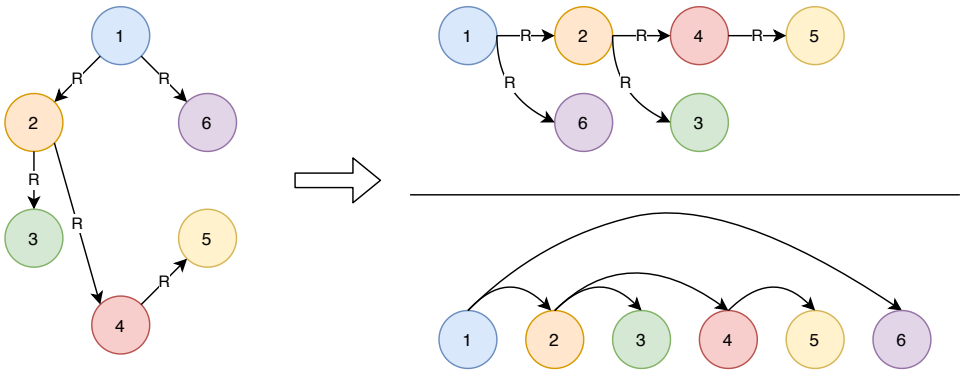
3.3 Performing the whole task: Our DepStructurer

The first design choice in addressing the task of simultaneously structuring and ordering a set of EDUs is whether the system should learn how to build dependency or constituency discourse trees (see Figure 1 (a)-(b) for corresponding examples). We decided to aim for dependency discourse structures for two key reasons. Not only have they been shown to be more general and expressive (Morey et al., 2018), but there are also readily available graph-based methods for learning and inference of dependency trees (Ma and Hovy, 2017) that when properly combined enable structure and ordering prediction to benefit from each other. However, since the only large-scale discourse treebank for training (MEGA-DT) contains constituency trees, we first convert them into dependency ones. For this, we follow the protocol of (Hayashi et al., 2016), which effectively resolves the ambiguity involved in converting multinuclear constituency units. Notice that we want dependency trees that fully specify the ordering for the EDUs, so our translation algorithm also labels each dependency arc with label - L or R, denoting whether the modifier node should be on the left (L) or on the right (R) of the head node in the linearized EDU sequence.

Once the training data is generated as a dependency treebank, our two-step solution for the task of structuring and ordering a set of EDUs can be applied. Notice that the same EDU embeddings $v_{1:n}$ are reused in both steps - for tree induction (Step 1 §3.3.1) and child ordering (Step 2 §3.3.2). These embeddings are generated by training a single EDU Encoder as described in §3.1.

3.3.1 Step 1: Compatibility Matrix and Initial Tree Induction

The first step of our solution learns how to build a discourse dependency tree for the input sequence of EDU embeddings $v_{1:n}$. Formally, this can be framed as learning a compatibility matrix (edge



a) Output of Step 1 b) Output of Step 2

Figure 2: Outputs of the two inference steps: (a) Initially induced Dependency Tree and (b) Final total ordering.

score tensor more precisely) $M \in \mathbb{R}^{n \times n \times 2}$, where the last dimension of l an entry i, j corresponds to the scores of the labels L and R for the edge from node i to node j. Similarly to (Ma and Hovy, 2017), each entry is computed as follows:

$$M_{i,j} = v_i^T W_1 v_j + W_2 v_i + W_3 v_j + b \quad (8)$$

where $W_1 \in \mathbb{R}^{d \times d \times 2}$, $W_2 \in \mathbb{R}^{d \times 2}$ and $W_3 \in \mathbb{R}^{d \times 2}, b \in \mathbb{R}^2$ are learnable bilinear, linear and bias terms. Once the tensor M is predicted, it is used for inferring an initial dependency structure.

Learning M: The objective is to maximize the probability of the correct tree structure $\mathbf{y}$:

$$P(\mathbf{y}|e_{1:n}, \theta) = \frac{\exp\left\{\sum_{(v_i,v_j,l)\in\mathbf{y}} M_{i,j,l}\right\}}{Z(e_{1:n}, \theta)} \quad (9)$$

where

$$Z(e_{1:n}, \theta) = \sum_{\mathbf{y}\in T(e_{1:n})} \exp\left\{\sum_{(v_i,v_j,l)\in\mathbf{y}} M_{i,j,l}\right\} \quad (10)$$

with $T(e_{1:n})$ denoting all possible trees from a set of EDUs $e_{1:n}$. Since the number $|T(e_{1:n})|$ of possible trees grows exponentially with the number of EDUs, we need an efficient way of computing $Z(e_{1:n}, \theta)$. Following (Koo et al., 2007), we achieve this goal by first computing the weighted adjacency matrix $A(M) \in \mathbb{R}^{n \times n \times 2}$ for left-child and right-child edges:

$$A_{i,j,l} = \begin{cases} 0, & \text{if } i = j \\ \exp\{M_{i,j,l}\} & \text{otherwise} \end{cases} \quad (11)$$

as well as the root scores for each node:

$$r_i(v) = \exp\{MLP(v_i)\} \quad (12)$$

Then, the weight of the correct dependency structure $\mathbf{y}$ is defined as

$$\psi(\mathbf{y}, \theta) = r_{root(\mathbf{y})}(v) \prod_{i,j,l\in\mathbf{y}} A_{i,j,l} \quad (13)$$

where $root(\mathbf{y})$ is the child of the root node in the dependency tree. We then compute the Laplacian matrix L of G:

$$L_{i,j} = \begin{cases} \sum_{i'=1}^{n} \sum_{l=1}^{2} A_{i',j,l}, & \text{if } i = j \\ \sum_{l=1}^{2} -A_{i,j,l} & \text{otherwise} \end{cases} \quad (14)$$

and replace its first row by $r(v)$:

$$\hat{L}_{i,j} = \begin{cases} r_i(v), & \text{if } i = 1 \\ L_{i,j} & \text{otherwise} \end{cases} \quad (15)$$

It can be shown (Koo et al., 2007) that the determinant of $\hat{L}$ is in fact equal to the normalizing constant that we need:

$$Z(e_{1:n}, \theta) = |\hat{L}| \quad (16)$$

which takes $O(n^3)$ time to compute. Hence, the loss for tree construction derived from eq. 9 can be computed efficiently:

$$l_{tree}(\theta) = -\log \psi(\mathbf{y}, \theta) + \log Z(e_{1:n}, \theta) \quad (17)$$

Inference of the initial tree structure: The learned model is applied to the input sequence of EDU embeddings $v_{1:n}$. Then, using the predicted compatibility matrix M, the highest-weighting tree structure can be constructed by the Chu-Liu-Edmonds algorithm (Edmonds, 1967), with the root being the node with highest root score r_i (eq. 12). Figure 2 (a) shows a sample output of this process.

3.3.2 Step 2: Ordering Children

The key limitation of Step 1 is that some nodes in the resulting dependency tree can have multiple left or right children, which makes their relative order unrecoverable from the basic tree structure. For instance, this is the case for nodes 1 and 2 in Figure 2 (a), both of which have two left children (outgoing edges labeled by L). To address this issue, in Step 2 for every node $s_i \in s_{1:n}$ that has $k > 1$ left or right children $s_{i_1}, ..., s_{i_k}$, we train a pointer network that predicts the correct order of children on each side - in the same way as described in § 3.2, but specifically trained on groups of children in MEGA-DT. Figure 2 (b)(top) illustrates an output of the Pointer network applied to plain dependency structure in Figure 2 (a), from which the final EDU ordering 2 (b)(bottom) is decoded as follows.

Algorithm 1: PredictEduOrder

```
   Data: Root
   Result: The ordering of elements of V
1  ordering = []
2  ordChildren = PtrNet(Root.leftChildren)
3  for child in ordChildren do
4  |   ordering.extend(PredictEduOrder(child))
5  end
6  ordering.append(Root)
7  ordChildren = PtrNet(Root.rightChildren)
8  for child in ordChildren do
9  |   ordering.extend(PredictEduOrder(child))
10 end
```

Inference of final ordering: The pseudocode for predicting the final ordering is provided in Algorithm 1. The ordering is built recursively bottom-up - at each step, given the ordering of all left and right subtrees (recursive calls in lines 4, 9), the ordering is obtained by concatenating, in the order predicted by Pointer network (lines 2, 7), the orderings of those subtrees, together with the current root node (line 6). Specifically, the children are ordered according to their root node; for example in Figure 2(b)(top), when deciding the order for child subtrees rooted at nodes 2,6 for the node 1, the pointer network orders them using the embeddings for those nodes.

3.4 Baselines for Ordering and Full Task

Language model decoding (LMD): greedily predicts the linear EDU ordering. The next EDU at each timestep is the one maximizing the length normalized language modelling objective from ALBERT.

Unsupervised tree induction (UTI): computes the compatibility matrix M using cosine similarity between the means of ALBERT embeddings for each EDU. The label for dependency (left vs. right child) is chosen randomly, while dependent orders for nodes with multiple children are chosen according to above ordering baseline LMD.

Tree Induction (TI+LMD): being an ablation for our main model, this baseline only learns to induce the tree structure in the same way as DepStructurer, but orders the children as in LMD, without performing supervised leaf ordering.

4 Experiments

4.1 The MEGA-DT Dataset

Our evaluation relies on MEGA-DT, a discourse treebank generated by distant supervision from the Yelp'13 corpus of customer reviews (Tang et al., 2015), according to the method presented by Huber and Carenini (2020). The high-quality of MEGA-DT trees has been certified in experiments in inter-domain discourse parsing similar to the ones described in (Huber and Carenini, 2020). In practice, their approach for generating the discourse trees for a set of documents can be applied to any other genre. If the required sentiment annotation is not naturally available (like star ratings for customer reviews), it can be obtained from an off-the-shelf sentiment analyzer. We train all models on 100k and 215k subsets of MEGA-DT, and use 7.5k documents for development and 15k for testing. Due to memory requirements induced by finetuning ALBERT, the training splits only contain documents with less than 35 EDUs, whereas to evaluate the performance on longer documents, the development and test sets contain respectively 2.5k and 5k of longer documents. The project GitHub repository provides the exact splits.

4.2 Evaluation Metrics

In all experiments, we assess the quality of the EDUs ordering and of their tree structure independently with two sets of corresponding metrics.

4.2.1 Information Ordering Metrics

Measuring the quality of information ordering is a challenging task because different metrics can be more or less appropriate depending on the num-

ber and the nature/granularity of the information units that are ordered. In accord with previous works, we first consider a set of simple metrics that essentially penalize the distance of an information unit from its correct position. These include Kendall's τ, Position Accuracy (POS) and Perfect Match Ratio (PMR). Then, we propose a new, more sophisticated metric, which is arguably much more appropriate for longer sequences of relatively short information units (i.e., sequences of EDUs of long multisentential text). This metric, that we call Blocked Kendall'τ rewards a correctly ordered sub-sequence even if its location is shifted as a single block.

Kendall's τ: a metric of rank correlation, widely used for information ordering evaluation; found to correlate with human judgement (Lapata, 2006). It is computed as follows:

$$\frac{1}{|D|}\sum_{o_i \in D} \tau_{\hat{o}_i} \tag{18}$$

where

$$\tau_{\hat{o}_i} = 1 - 2 * \frac{\text{\# of transpositions}}{\binom{n}{2}} \tag{19}$$

Position Accuracy (POS) computes the average proportion of EDUs that are in their correct absolute position according to the gold ordering.

$$\frac{1}{|D|}\sum_{o_i \in D} \frac{count(\hat{o}_i = o_i)}{length(o_i)} \tag{20}$$

Perfect Match Ratio (PMR) is the strictest metric, measuring the proportion of documents where positions of all EDUs are predicted correctly.

$$\frac{1}{|D|}\sum_{o_i \in D} 1(\hat{o}_i = o_i) \tag{21}$$

The new metric Blocked Kendall's τ: All metrics from previous work simply penalize the distance of an information unit from its correct position. However, ideally, a good metric for information ordering should also capture how well semantically close units are clustered together. This aspect is even more critical when ordering discourse units of long documents - oftentimes, paragraphs or groups of sentences are largely independent in their meaning from other parts of text, so as long as a paragraph's subset of EDUs is ordered correctly, placing it in a different position should not be penalized harshly. As a short example, given the correct ordering o_c [1, 2, 3, 4, 5], all aforementioned metrics would give a low score to the predicted ordering o_p [3, 4, 5, 1, 2] - zero for PMR and POS, and -0.2 for Kendall's τ. Yet, since the blocks [1, 2] and [3, 4, 5] are preserved in o_p, it makes sense to penalize this ordering for only one transposition, and not for twelve like Kendall's τ does. Arguably, these blocks of EDUs are likely to be much more coherent and interpretable than random sequences.

Therefore, we propose a modification for Kendall's τ that treats the correctly ordered blocks as single units. For the example above with $n = 5$, we first merge the correct blocks into single units (indexed by the first EDU in the block), so $[3, 4, 5, 1, 2] \rightarrow [3, 1]$, and compute the Kendall's τ on the resulting reduced sequence:

$$\textit{Block } \tau_{\hat{o}_i} = 1 - 2\frac{\text{\# block transpositions}}{\binom{n}{2}} \tag{22}$$

The number of transpositions can be at least zero (if the sequence is perfectly ordered) and at most $\binom{n}{2}$, if the sequence is in reversed order. Thus, Blocked Kendall's τ has the same range $[-1, 1]$ and is lower bounded by the standard Kendall's τ, with the key advantage of rewarding correct blocks of EDUs. We also note that our proposed measure and the standard Kendall's τ are not metrics in mathematical sense, as they both give a score of 1 to perfectly ordered sequences.

4.2.2 Tree Structure Metrics

UAS and LAS: Unlabelled and labelled attachment scores are the most commonly used measures for evaluation of dependency parsers:

$$\text{UAS} = \frac{\{e | e \in E_G \cap E_P\}}{|V|} \tag{23}$$

$$\text{LAS} = \frac{\{e | l_G(e) = l_P(e), e \in E_G \cap E_P\}}{|V|} \tag{24}$$

where V is the set of EDUs, E_G, E_P are the sets of gold and predicted edges, and $l_G(e)$ is the label of edge e in G.

5 Quantitative and Qualitative Results

Results are presented in Table 1 for the full test set (upper sub-table) and its longer (> 35 EDUs) document subset (lower-sub-table). Remarkably, the DepStructurer (§3.3) dominates other approaches on the new ordering metric (Blocked Kendall's τ),

Approach	New ordering metric	Tree structure		Previous ordering metrics		
	Blocked τ	UAS	LAS	Kendall's τ	POS	PMR
Full test set						
LM Decoding	8.7	×	×	-1.3	8.4	1.86
Unsup Tree Induction (UTI)	10.7	13.1	9.27	0.3	9.3	2.61
Tree Induction (TI+LMD) (100k)	41.7	24.5	22.9	20.0	16.9	7.36
Tree Induction (TI+LMD) (215k)	45.6	**25.9**	**24.3**	21.2	17.5	7.76
Pointer Network (100K)	38.2	×	×	29.4	19.9	6.89
Pointer Network (215K)	40.4	×	×	**31.3**	20.7	7.22
DepStructurer (100K)	48.7	24.3	22.7	28.8	20.0	8.90
DepStructurer (215K)	**52.7**	25.8	24.2	30.7	**21.0**	**9.35**
Long documents only (> 35 EDUs)						
LM Decoding	2.4	×	×	-1.7	2.0	0
Unsup Tree Induction (UTI)	4.5	3.41	2.22	0.0	2.07	0
Tree Induction (TI+LMD) (100k)	21.2	12.4	11.5	5.0	2.8	0
Tree Induction (TI+LMD) (215k)	25.1	**13.6**	**12.7**	5.5	3.0	0
Pointer Network (100K)	21.9	×	×	16.6	4.5	0
Pointer Network (215K)	24.1	×	×	**18.3**	**4.84**	0
DepStructurer (100K)	27.5	12.0	11.1	11.7	3.55	0
DepStructurer (215K)	**31.5**	13.4	12.5	12.3	3.51	0

Table 1: Evaluation results on full test set (15k documents) and its long-document subset (5k documents), with best results per subtable highlighted in **bold**. The entries marked as (×) signify that these metrics cannot be computed for the corresponding models, since they do not induce document tree structures.

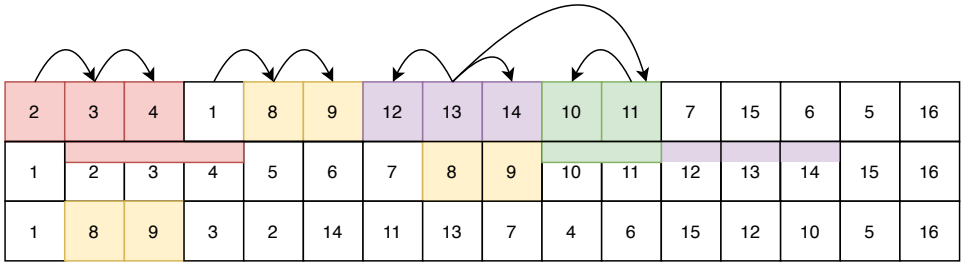


Figure 3: Ordering produced by DepStructurer (top row) and Pointer (bottom row); Gold ordering in middle row.

and surprisingly, our TI+LMD baseline also outperforms the Pointer Network on the full test set and has the performance similar to it on the long-document subset. In contrast, results are mixed for ordering metrics from previous work (last column), which as we have argued in §4.2.1 are however less appropriate for our text structuring task. Interestingly, all trainable models (Pointer Networks §3.2, our DepStructurer §3.3 and TI+LMD §3.4) benefit from more training data (100K → 215K), with equal or even bigger absolute gains for the DepStructurer, especially on the new metric. This validates the quality of the MEGA-DT treebank and suggests that training on larger corpora could increase the performance even further.

Focusing on the performance of tree induction systems, our DepStructurer outperforms the unsupervised model (UTI) by a wide margin and has nearly identical performance with TI+LMD model, indicating that a trainable tree induction model is essential to obtain much more accurate trees.

Lastly, among the unsupervised models, UTI outperforms LM across all metrics. This suggests that even without training, forcing a model to generate a tree structure is by itself a useful inductive bias.

To highlight the strengths and potential weaknesses of our solution and new metric, we analyze the output of the DepStructurer and Pointer models for two medium-length illustrative sample documents with 16 and 14 EDUs respectively (see Figures 3 and 4). In each figure, the top row indicates the ordering output of the DepStructurer, the middle row is the gold (i.e., correct) ordering, and the bottom is the Pointer's output. We color-coded the blocks that each model predicted correctly, with the highlights in the middle gold ordering denoting whether the top or bottom model predicted that block correctly. Additionally, for both exam-

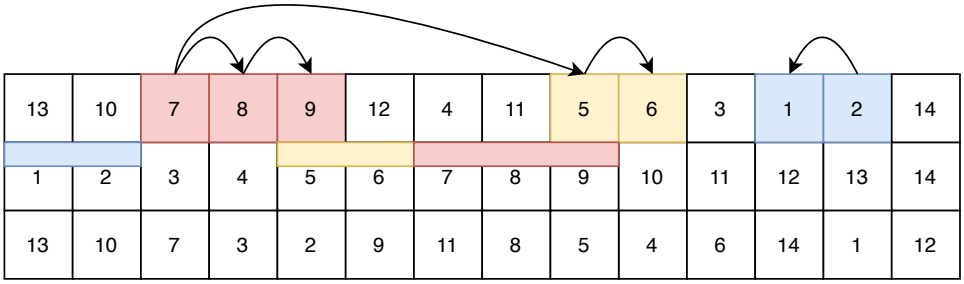


Figure 4: Example illustrating benefits of new metric.

ples, on the top of the DepStructurer ordering, we show the predicted tree dependency edges within the blocks. The main structural benefit of the DepStructurer can be clearly seen in the Figure 3 - the adjacent EDUs tend to form subtrees, the nodes of which the model learns to put close together. In the case of the Pointer model, however, even though it was able to infer a reasonable approximate ordering - with EDUs 1, 3, 2 and EDUs 15, 12, 16 being placed respectively at the beginning/end of the sequence, it failed to arrange them properly in coherent blocks. In Figure 4, we can see an example where the DepStructurer scores in the standard and Blocked Kendall's τ are very different: -36.3 vs. 34.1; while they are the same for the pointer model -9.9. This example clearly illustrates the benefit of our new metric for text structuring. While both models made poor predictions with respect to the distance of each EDU to its correct position, our DepStructurer arguably learned a much more coherent document structure by better grouping related information, which is reflected in the Blocked metric, but is ignored by the standard Kendall's τ.

6 Conclusions and Future work

By proposing the domain-independent task of structuring and ordering a set of EDUs, we aim to stimulate more general and data-driven approaches for text structuring. The solution we have developed for such task combines neural dependency tree induction with pointer networks, which are both trainable on large discourse treebanks. Since existing text ordering metrics are not capturing key aspects of text structuring, we have also proposed a new metric that is arguably much more suitable for the task. In a series of experiments, complemented by qualitative error analysis, we have shown that our solution delivers top performance and represents a promising initial framework for further developments. Fruitful directions for future work include: **(1)** Exploring more recent techniques for tree induction, such as pointer-based and higher-order dependency parsing. **(2)** Integrating our approach into existing long-document data-to-text NLG pipelines such as Puduppully et al. (2019), to explore the benefits of content structuring pre-training for data-to-text applications. **(3)** Verifying the validity of our proposed measure for ordering textual units of long documents (i.e. correlation with human judgement), as well as exploring further metrics for text structuring. **(4)** Extending our approach to fully-labelled RST discourse trees involving nuclearity and relation annotations, which can be obtained from state-of-the-art RST discourse parsers.

References

Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy. Association for Computational Linguistics.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 128–135, New York, NY, USA. Association for Computing Machinery.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349, Brussels, Belgium. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *JOURNAL OF RESEARCH of the National Bureau of Stan-*

dards - B. Mathematics and Mathematical Physics, 71B:233–240.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. Evaluating discourse in structured text representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 646–653, Florence, Italy. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2017. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61.

Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2016. Empirical comparison of dependency conversions for RST discourse trees. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 128–136, Los Angeles. Association for Computational Linguistics.

Patrick Huber and Giuseppe Carenini. 2020. Mega rst discourse treebanks with structure and nuclearity from scalable distant sentiment supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2102–2113, Florence, Italy. Association for Computational Linguistics.

Dan Jurafsky and James H Martin. 2014. *Speech and language processing*, volume 3. Pearson London.

Hamid Karimi and Jiliang Tang. 2019. Learning hierarchical discourse-level structure for fake news detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3432–3442, Minneapolis, Minnesota. Association for Computational Linguistics.

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1745–1755, Minneapolis, Minnesota. Association for Computational Linguistics.

Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2016. Sentence ordering and coherence modeling using recurrent neural networks. In *AAAI*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xuezhe Ma and Eduard Hovy. 2017. Neural probabilistic model for non-projective MST parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 59–69, Taipei, Taiwan. Asian Federation of Natural Language Processing.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2018. A dependency perspective on RST discourse parsing and evaluation. *Computational Linguistics*, 44(2):197–235.

Ani Nenkova and Kathleen McKeown. 2012. *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA.

Vassilis Plachouras, Charese Smiley, Hiroko Bretz, Ola Taylor, Jochen L. Leidner, Dezhao Song, and Frank Schilder. 2016. Interacting with financial data using natural language. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 1121–1124, New York, NY, USA. Association for Computing Machinery.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *AAAI*.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, USA.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3257–3268, Hong Kong, China. 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. *Journal of Machine Learning Research*, 15(56):1929–1958.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Tianming Wang and Xiaojun Wan. 2019. Hierarchical attention networks for sentence ordering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:7184–7191.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2019. Discourse level factors for sentence deletion in text simplification.

A Hyperparameters and training setup

For the Pointer Model §3.2, similarly to (Cui et al., 2018), the hidden state size in the decoder and transformer EDU encoder is 512, and beam size is 64. Also, as in Cui et al. (2018), the 4-layer Transformer has 8 attention heads. For the Dependency Model §3.3, the edge prediction weights have $d = 512$, and we choose the highest-scoring tree among the top-5 root classifier predictions during inference. The 768-dimensional outputs of ALBERT are transformed with a dense layer to match the dimensionality of EDU encoder. We use AdamW optimizer (Loshchilov and Hutter, 2019) with default weight decay 0.01 and learning rate 0.001, and clip gradient norm at 0.2. The learning rate scheduling rule as in (Vaswani et al., 2017) has 4000 warm-up steps. We apply word dropout (Srivastava et al., 2014) to outputs of ALBERT and of the contextual EDU encoder. We tune dropout value using 15k training subset, selecting among [0, 0.05, 0.15, 0.3], with best values 0.15 for Pointer and 0 for the Dependency Model. All models are trained using early stopping if validation loss did not decrease for three epochs. As only 1% of EDUs have length $>$ 20 word tokens, we clip each EDU's size at 50 ALBERT tokenizer tokens (since it keeps spaces). Batch size for all models is 2 - the highest that could fit into a single GTX 1080 Ti GPU with 11 GB of memory.

B EDU Ordering Examples

See the next page.

Dependency:

2: the lechon special on saturdays tasted 3: like it was premade. 4: the ``crispy`` part of the pork belly was almost gooey. 1: i would actually go for 2 1/2 stars. 8: for $2.00, you get 5 mini half, 9: that are great! 12: being a true filipino, i like my lumpia with a vinegar sauce. 13: if you ask the cashier, for a vinegar sauce, 14: they have a white vinegar, with some onions in it. 10: they give you a sweet and sour sauce on the side, 11: which i do n't think goes well with it. 7: the gem was the shanghai. 15: it was ok, better then then the sweet and sour. 6: the pancit was good, but heavy on the vegetables. 5: the meat itself tasted good, although better with some kikkoman shoyu. 16: overall, a descent find.

Gold:

1: i would actually go for 2 1/2 stars. 2: the lechon special on saturdays tasted 3: like it was premade . 4: the ``crispy`` part of the pork belly was almost gooey. 5: the meat itself tasted good , although better with some kikkoman shoyu. 6: the pancit was good, but heavy on the vegetables. 7: the gem was the shanghai. 8: for $2.00 , you get 5 mini half, 9: that are great! 10: they give you a sweet and sour sauce on the side, 11: which i don't think goes well with it. 12: being a true filipino , i like my lumpia with a vinegar sauce. 13: if you ask the cashier, for a vinegar sauce, 14: they have a white vinegar, with some onions in it. 15: it was ok , better then then the sweet and sour. 16: overall, a descent find.

Pointer:

1: i would actually go for 2 1/2 stars. 8: for $2.00 , you get 5 mini half , 9: that are great! 3: like it was premade. 2: the lechon special on saturdays tasted 14: they have a white vinegar, with some onions in it. 11: which i don't think goes well with it. 13: if you ask the cashier, for a vinegar sauce, 7: the gem was the shanghai. 4: the ``crispy`` part of the pork belly was almost gooey. 6: the pancit was good, but heavy on the vegetables. 15: it was ok, better then then the sweet and sour. 12: being a true filipino, i like my lumpia with a vinegar sauce. 10: they give you a sweet and sour sauce on the side, 5: the meat itself tasted good, although better with some kikkoman shoyu. 16: overall, a descent find.

Figure 5: Example from Figure 3 in the paper

Dependency:

13: i simply love their gyros! 10: it is set up like sauce 7: the food is cooked fresh 8: for you 9: so there will be a short wait. 12: and they bring the food to you. 4: the interior is cutesy and bright 11: where you order at the cashier area 5: while upbeat music is playing. 6: they have a small outdoor seating area and some booths and tables inside. 3: it's tucked away in a strip plaza shockingly! 1: i hope more people are frequenting this place 2: since i was last there. 14: it's relatively quick but always fresh and inexpensive!

Gold:

1: i hope more people are frequenting this place 2: since i was last there. 3: it's tucked away in a strip plaza shockingly! 4: the interior is cutesy and bright 5: while upbeat music is playing. 6: they have a small outdoor seating area and some booths and tables inside. 7: the food is cooked fresh 8: for you 9: so there will be a short wait. 10: it is set up like sauce 11: where you order at the cashier area, 12: and they bring the food to you. 13: i simply love their gyros! 14: it's relatively quick but always fresh and inexpensive!

Pointer:

13: i simply love their gyros! 10: it is set up like sauce 7: the food is cooked fresh 3: it's tucked away in a strip plaza shockingly! 2: since i was last there. 9: so there will be a short wait. 11: where you order at the cashier area 8: for you, 5: while upbeat music is playing. 4: the interior is cutesy and bright 6: they have a small outdoor seating area and some booths and tables inside. 14: it's relatively quick but always fresh and inexpensive! 1: i hope more people are frequenting this place 12: and they bring the food to you.

Figure 6: Example from Figure 4 in the paper