University of Edinburgh's Submission to the Document-level Generation and Translation Shared Task

Ratish Puduppully * and **Jonathan Mallinson** * and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
r.puduppully@sms.ed.ac.uk J.Mallinson@ed.ac.uk mlap@inf.ed.ac.uk

Abstract

The University of Edinburgh participated in all six tracks: NLG, MT, and MT+NLG with both English and German as targeted languages. For the NLG track, we submitted a multilingual system based on the Content Selection and Planning model of Puduppully et al. (2019). For the MT track, we submitted Transformer-based Neural Machine Translation models, where out-of-domain parallel data was augmented with in-domain data extracted from monolingual corpora. Our MT+NLG systems disregard the structured input data and instead rely exclusively on the source summaries.

1 Track 1/2: Natural Language Generation

The Natural Language Generation (NLG) track revolved around systems that take structured data in the form of tabular data from a basketball game as input, and generate a summary of this game in the target language. We entered one multilingual system which outputs summaries in both English and German. A multilingual model allows us to overcome the limited amount of German training data.

We adopted the content selection and planning approach of Puduppully et al. (2019), made extensions to the model and parameterized the decoder with a language tag, indicating the target language. The training was done using the full ROTOWIRE English dataset and the ROTOWIRE English-German dataset. We first explain the approach of Puduppully et al. (2019), describe the extensions to their model and show how language tags can be added to the decoder to indicate the target language.

1.1 The Content Selection and Planning Approach of Puduppully et al. (2019)

Puduppully et al. (2019) model $p(y|r)$ as the joint probability of text y and content plan z, given input r. They further decompose $p(y, z|r)$ into $p(z|r)$, a content selection and planning phase, and $p(y|r, z)$, a text generation phase:

$$p(y|r) = \sum_{z} p(y, z|r) = \sum_{z} p(z|r)p(y|r, z)$$

Given input records, probability $p(z|r)$ is modeled using Pointer Networks (Vinyals et al., 2015). The probability of output text y conditioned on previously generated content plan z and input table r is modeled as follows:

$$p(y|r, z) = \prod_{t=1}^{|y|} p(y_t|y_{<t}, z, r)$$

where $y_{<t} = y_1 \dots y_{t-1}$. They use an encoder-decoder architecture with an attention mechanism to compute $p(y|r, z)$. The architecture is shown in Figure 1.

The content plan z is encoded into $\{\mathbf{e}_k\}_{k=1}^{|z|}$ using a bidirectional LSTM. Because the content plan is a sequence of input records, they directly feed the corresponding content selected record vectors $\{\mathbf{r}_j^{cs}\}_{j=1}^{|r|}$ as input to the LSTM units, which share the record encoder with the first stage. For details of the content selection stage, please refer Puduppully et al. (2019).

The text decoder is also based on a recurrent neural network with LSTM units. The decoder is initialized with the hidden states of the final step in the encoder. At decoding step t, the input of the LSTM unit is the embedding of the previously

*Ratish worked on Tracks 1/2 and Jonathan on Tracks 3/4/5/6.

Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019), pages 268–272
Hong Kong, China, November 4, 2019.
www.aclweb.org/anthology/D19-56%2d

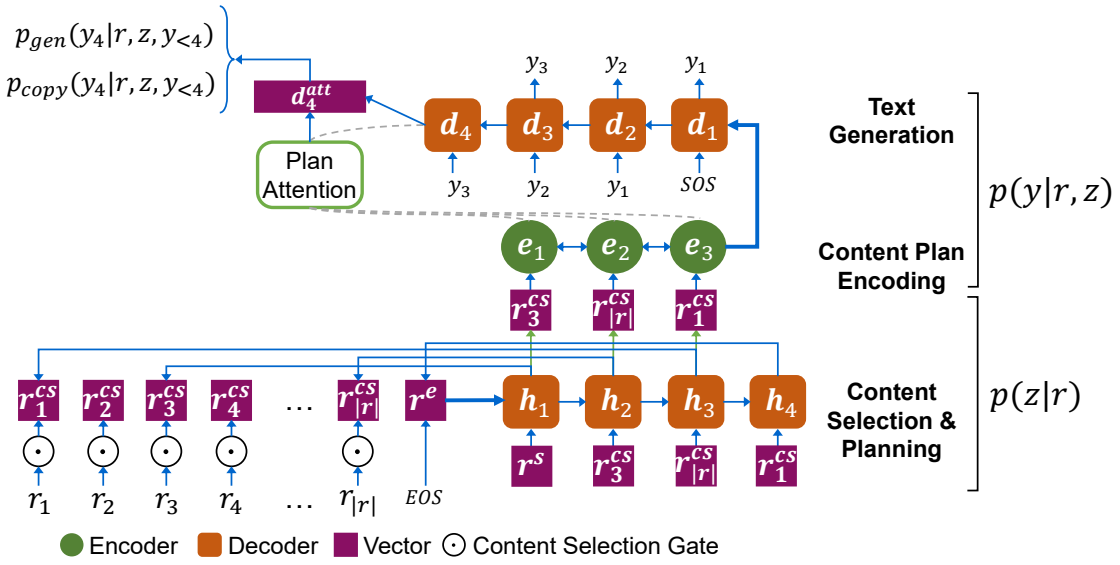

Figure 1: Generation model with content selection and planning. The text is generated conditioned on the input content plan. At any time step, output token is generated from vocabulary or copied from the content plan.

predicted word y_{t-1}. Let $\mathbf{d}_t$ be the hidden state of the t-th LSTM unit. The probability of predicting y_t from the output vocabulary is computed via:

$$\beta_{t,k} \propto \exp(\mathbf{d}_t^\intercal \mathbf{W}_b \mathbf{e}_k) \quad (1)$$

$$\mathbf{q}_t = \sum_k \beta_{t,k} \mathbf{e}_k$$

$$\mathbf{d}_t^{att} = \tanh(\mathbf{W}_d[\mathbf{d}_t; \mathbf{q}_t])$$

$$p_{gen}(y_t|y_{<t}, z, r) = \text{softmax}_{y_t}(\mathbf{W}_y \mathbf{d}_t^{att} + \mathbf{b}_y) \quad (2)$$

where $\sum_k \beta_{t,k} = 1$, $\mathbf{W}_b \in \mathbb{R}^{n \times n}$, $\mathbf{W}_d \in \mathbb{R}^{n \times 2n}$, $\mathbf{W}_y \in \mathbb{R}^{n \times |\mathcal{V}_y|}$, $\mathbf{b}_y \in \mathbb{R}^{|\mathcal{V}_y|}$ are parameters, and $|\mathcal{V}_y|$ is the output vocabulary size.

They further augment the decoder with a copy mechanism, allowing the ability to copy words directly from the *value* portions of records in the content plan (i.e., $\{z_k\}_{k=1}^{|z|}$). They experimented with joint (Gu et al., 2016) and conditional copy methods (Gulcehre et al., 2016). Specifically, they introduce a variable $u_t \in \{0, 1\}$ for each time step to indicate whether the predicted token y_t is copied ($u_t = 1$) or not ($u_t = 0$). The probability of generating y_t is computed by:

$$p(y_t|y_{<t}, z, r) = \sum_{u_t \in \{0,1\}} p(y_t, u_t|y_{<t}, z, r)$$

where u_t is marginalized out.

1.2 Copying from Table and Plan

We extended the copy mechanism further such that u_t can take three values: y_t is generated from the vocabulary ($u_t = 0$), y_t is copied from the content plan ($u_t = 1$) and y_t is copied from the table ($u_t = 2$).

Conditional Copy The variable u_t is first computed as a switch gate, and then is used to obtain the output probability:

$$p(u_t|y_{<t}, z, r) = \text{softmax}(\mathbf{w}_u \cdot \mathbf{d}_t^{att} + b_u)$$

$$\alpha_{t,j} \propto \exp(\mathbf{d_t}^\intercal \mathbf{W}_c \mathbf{r}_j^{cs}) \quad (3)$$

$$p(y_t, u_t|y_{<t}, z, r) =$$

$$\begin{cases} p(u_t|y_{<t}, z, r) \sum_{y_t \leftarrow z_k} \beta_{t,k} & u_t = 1 \\ p(u_t|y_{<t}, z, r) \sum_k \beta_{t,k} \sum_{y_t \leftarrow r_j, j \in \gamma_k} \alpha_{t,j} & u_t = 2 \\ p(u_t|y_{<t}, z, r) p_{gen}(y_t|y_{<t}, z, r) & u_t = 0 \end{cases}$$

where $\sum_{j \in \gamma_k} \alpha_{t,j} = 1$. $y_t \leftarrow z_k$ indicates that y_t can be copied from z_k, $y_t \leftarrow r_j$ indicates that y_t can be copied from r_j. γ_k indicates records in table corresponding to the kth record in plan, for example: if k is 'PTS' value of player *Jeff Teague*, then γ_k corresponds to all the records for the entity *Jeff Teague* in the table including 'PTS', 'REB', 'NAME1', 'NAME2' etc. $\beta_{t,k}$ and $p_{gen}(y_t|y_{<t}, z, r)$ are computed as in Equations (1)–(2), and $\mathbf{w}_u \in \mathbb{R}^{3 \times n}, b_u \in \mathbb{R}^3$ are parameters.

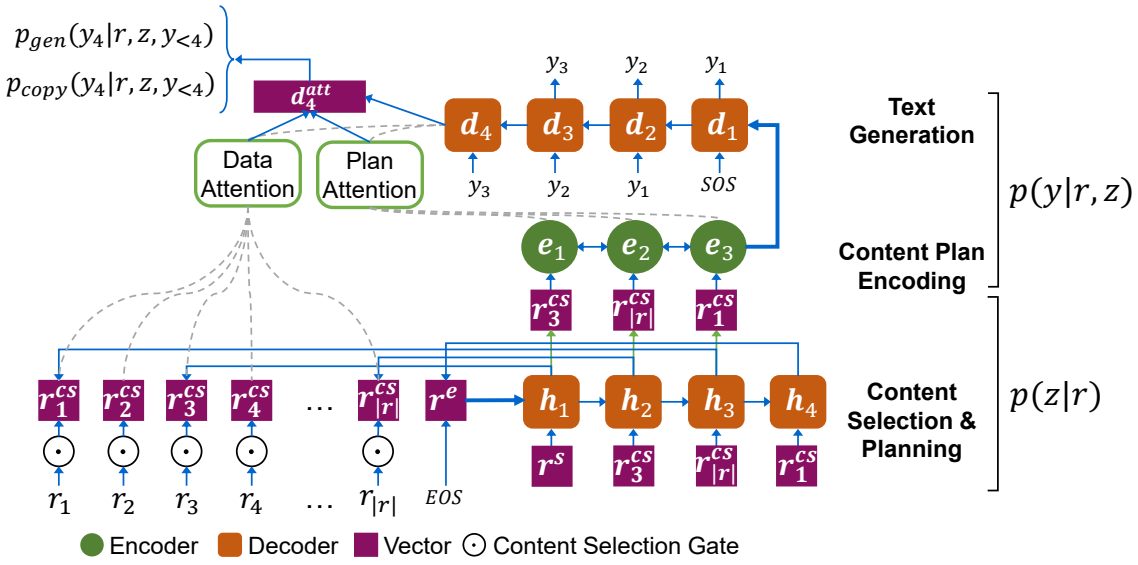

Figure 2: Generation model with content selection and planning and attention over table and content plan. The text is generated conditioned on the content plan and the table. At any time step, output token is generated from vocabulary, copied from the content plan or copied from input table.

1.3 Attending to the Table and Content Plan

The output text is generated by attending to both the content plan and the input table (See Figure 2.)

$$\delta_{t,j} \propto \exp(\mathbf{d}_t^\intercal \mathbf{W}_c \mathbf{r}_j^{CS}) \quad (4)$$

$$\mathbf{s}_t = \sum_j \delta_{t,j} \mathbf{r}_j^{CS}$$

$$\mathbf{d}_t^{att} = \tanh(\mathbf{W}_d[\mathbf{d}_t; \mathbf{q}_t; \mathbf{s}_t])$$

$$p_{gen}(y_t|y_{<t}, z, r) = \text{softmax}_{y_t}(\mathbf{W}_y \mathbf{d}_t^{att} + \mathbf{b}_y) \quad (5)$$

where $\sum_j \delta_{t,j} = 1$, $\mathbf{W}_c \in \mathbb{R}^{n \times n}$, $\mathbf{W}_d \in \mathbb{R}^{n \times 3n}$, $\mathbf{W}_y \in \mathbb{R}^{n \times |\mathcal{V}_y|}$, $\mathbf{b}_y \in \mathbb{R}^{|\mathcal{V}_y|}$ are parameters, and $|\mathcal{V}_y|$ is the output vocabulary size.

1.4 Feature for Team Points and Ranking of Player Points

Upon inspection of the ROTOWIRE game summaries in the development set, we observed that the summaries often describe the statistics of the winning team followed by the statistics of the losing team. The highest ranked players of either team are also often described in sequence in the summaries. Currently, we rely on the word embeddings of the team and player points to help the model disambiguate the winning from the losing team and to learn the relative performances of the players. We hypothesize that explicitly providing information about the relative performance of players and teams should make the learning easier.

We thus experimented with a feature for the winning/losing team and the ranking of player points within a team. Specifically, we added a binary feature for team records: *win* for each record in the winning team, *loss* for each record in the losing team. We further rank players in a team on the basis of their points and we add a feature indicating their rank in the team. For instance, *Kyle Lowry* scored the highest number of points in the home team and we add feature *hometeam-0* to each of his records. Player *Jahlil Okafor* was the second highest scorer in the visiting team and we add the feature *visteam-1* to each of his records and so on.

1.5 Training a Single Multilingual Model

We trained a single model for English and German data-to-text with a common BPE (Sennrich et al., 2015b) vocabulary of 2000 symbols for the output summaries. Player names and values of records in summaries were not BPEd. The target text was prefixed with token indicating the language of output 'EN' or 'DE'. During inference, we forced the model to generate output in the desired language.

1.6 Dataset

We made use of the full ROTOWIRE English dataset of Wiseman et al. (2017) and the German dataset provided as part of the shared task. The statistics of the dataset are given in Table 1.

	Train	Dev	Test
English	3398	727	728
German	242	240	241

Table 1: Count of examples in Training, Development and Test sections of English and German dataset.

Model	RG	CS		CO	BLEU
	P%	P%	R%	DLD%	
EN	91.41	30.91	64.13	21.72	17.01
DE	70.23	23.40	41.83	16.08	10.95

Table 2: Automatic evaluation for track 1/2 on the ROTOWIRE test set using record generation (RG) precision, content selection (CS) precision and recall, content ordering (CO) in normalized Damerau-Levenshtein distance, and BLEU.

1.7 Results

Table 2 shows our results for English and German datasets on the Test set as provided by the shared task organizers.

2 Track 3/4 : Machine Translation

The Machine Translation (MT) track revolves arround systems that translate source summaries to the target language. Our submission takes advantage of existing state-of-the-art techniques in machine translation, including (1) transformer networks (Vaswani et al., 2017). (2) subword units (Sennrich et al., 2015b) and (3) the inclusion of in-domain monolingual data used via back-translation (Sennrich et al., 2015a).

For our submission, we focus on finding in-domain basketball summary data from within general-purpose monolingual datasets. We develop several heuristics allowing us to extract millions of in-domain monolingual sentences, which are then back-translated and included within the training data. This additional monolingual data improves bleu scores between 5 and 7 points.

2.1 Data

The translation models were trained on both the ROTOWIRE English-German and all WMT19 parallel training data. A summary of the training data can be found in table 3. For ease of comparison to the NLG task, tokenization was done using the tokenizer provided by the shared task organizers. BPE was employed with a joint BPE subword vocabulary of 50k.

Dataset	Size
Europarl v9	18.39
Common Crawl corpus	24.00
News Commentary v14	3.38
Document-split Rapid corpus	14.01
Wikititles	13.05
ParaCrawl	162.64
ROTOWIRE EN-DE	0.033
Total	235.47

Table 3: Size (number of parallel training sentences) in 100,000 of the EN-DE training data.

2.1.1 In-Domain Parallel Data

Table 3 highlights the extremely limited amount of in-domain parallel training data used; ROTOWIRE English-German makes up only 0.001% the parallel training data. To ensure our translation system produces in-domain translation, we supplemented the parallel data with in-domain monolingual data. We used back-translation to translate clean monolingual data from the target language to the source language.

Finding in-domain data for basketball is not trivial, as there are no explicit basketball WMT19 monolingual training sets. Therefore, we extracted in-domain basketball data from the available general-purpose monolingual datasets.

We considered all documents within the News Crawl 2007-2018 dataset and included all sentences which appeared within a document where any of the following conditions were met: (1) Contains a player's name, as taken from the ROTOWIRE English-German training data; (2) Contains two team names; (3) the title contains the word *NBA*. For German, 1.1 million monolingual target sentences were collected, and for English, 4.32 million monolingual target sentences. These sentences were then back-translated via sampling (Edunov et al., 2018) and used to augment the parallel training data.

2.2 Model Description

For our submissions, we used the Transformer model as implemented within OpenNMT-py (Klein et al., 2017). Transformers are state-of-the-art NMT approaches which rely on multi-headed attention applied to both the source and target sentences. All experiments are performed with 6 encoder-decoder layers, with an embedding layer of size 512, a feed-forward layer size of 2048, and

	EN-DE	DE-EN
Monolingual	34.44	40.72
Parallel	28.65	33.48

Table 4: Track 3-6: ROTOWIRE dev set results, showing BLEU without monolingual data *Parallel* and with monolingual data *Monolingual*.

Model	RG	CS		CO	BLEU
	P%	P%	R%	DLD%	
EN-DE	81.01	77.32	78.49	62.21	36.85
DE-EN	91.40	78.99	63.04	51.73	41.15

Table 5: Automatic evaluation for track 3-6 on the ROTOWIRE test set using record generation (RG) precision, content selection (CS) precision and recall, content ordering (CO) in normalized Damerau-Levenshtein distance, and BLEU.

8 attentional heads. We set the batch size to 4096 tokens and maximum sentence length to 100 BPE subwords. Dropout and label smoothing were also both set to 0.1. All other settings were set their default values as specified in OpenNMT-py. Decoding was performed with a beam size of ~15, length penalty averaging, and the decoder was constrained to block repeating 4-grams. Model selection was done using the BLEU score on the development set.

2.3 Results

Results on the development set in Table 4 show that the inclusion of monolingual data leads to a significant increase in bleu (between 5 and 7 points). Table 5 shows test set results for both English and German target languages. The results were provided by the shared task organizers.

3 Track 5/6: MT + NLG

The MT + NLG track combines the previous tracks, models take in as input both the structured data and the summary in the source language and produce a summary in the target language as output. We chose to disregard the structured data and instead exclusively use the source summary, translating it to the target language. As such this submission to this track is a replication of our MT submission with results shown in Table 5.

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.

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

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.

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.