Sentence-Level Content Planning and Style Specification for Neural Text Generation

Xinyu Hua and **Lu Wang**
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
hua.x@husky.neu.edu luwang@ccs.neu.edu

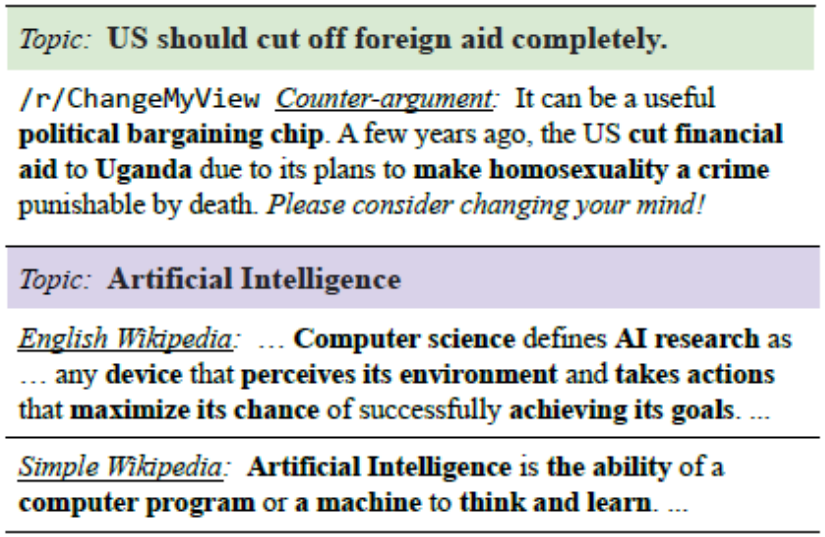
- Idea: three critical components: content selection, text planning, and surface realization, but all-in-one style neural generation models often produce outputs that are incoherent and unfaithful to input
- present an end-to end trained two-step generation model
- A sentence-level planner decoder decides on the keyphrases to cover as well as a desired language style, followed by a surface realization decoder that generates relevant and coherent text.

data

(1)ChangeMyView , each thread consists of an original post, followed by user replies with the intention to change the opinion of the OP user

(2)Wikipedia: Paragraph Generation for Normal and SimpleWikipedia

(3)AGENDA, for scientific paper abstract



FrameWork

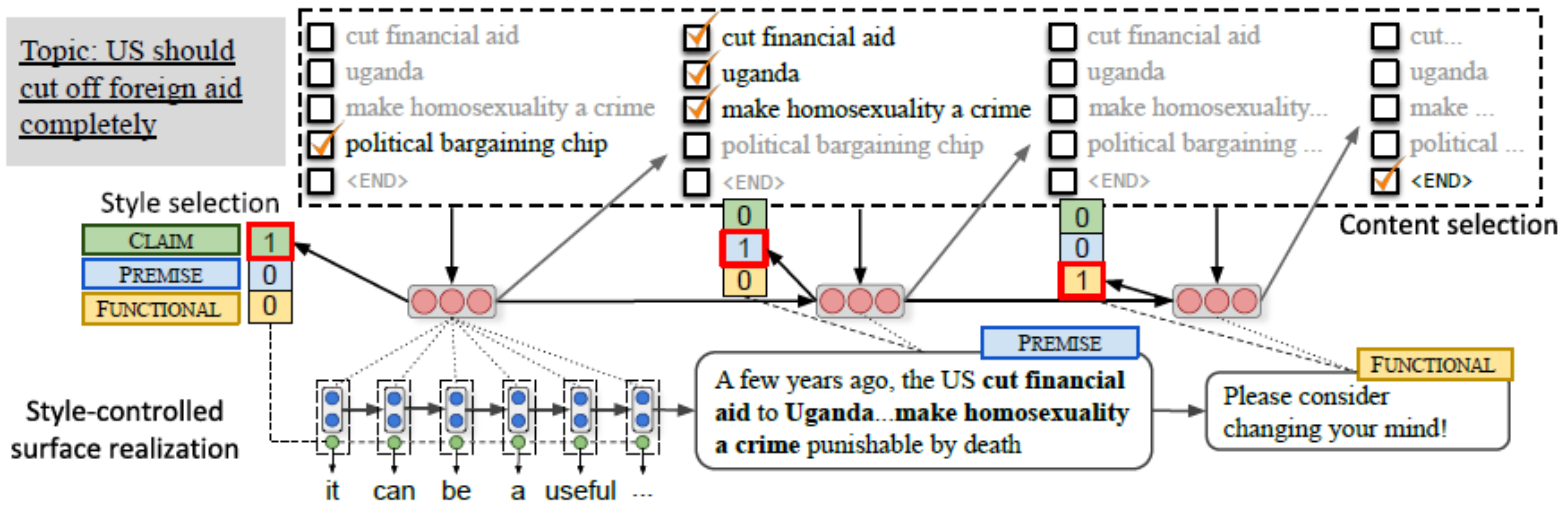


Figure 2: Overview of our framework. The LSTM content planning decoder (§ 3.2) first identifies a set of keyphrases from the memory bank conditional on previous selection history, based on which, a style is specified. During surface realization, the hidden states of the planning decoder and the predicted style encoding are fed into the realizer, which generates the final output (§ 3.3). Best viewed in color.

Model

- **Input Encoding**
- input text x is encoded via a bidirectional LSTM (biLSTM),
- To encode keyphrases in memory bank $\mathcal{M}$, first summing up all its words' embeddings for each keyphrase, then use a biLSTM to obtain $\boldsymbol{h}_k^e$, $k = 1,2, \dots, |\mathcal{M}|$

Model

- **Content Planning: Context-Aware Keyphrase Selection.**
- content planner selects a set of keyphrases from the memory bank $\mathcal{M}$ for each sentence. For each sentence indexed by j , predict a vector $v_j \in \mathbb{R}^{|\mathcal{M}|}$, where $v_{j,k} \in \{0,1\}$ indicating whether the k-th phrase is selected for the j-th sentence generation.
- utilize a sentence-level LSTM f, which consumes the summation embedding of selected keyphrases m_j, produces

$$s_j = f(s_{j-1}, m_j)$$

$$m_j = \sum_{k=1}^{|\mathcal{M}|} v_{j,k} h_k^e$$

Model

- **Content Planning: Context-Aware Keyphrase Selection.**
- keyphrases that have already been utilized many times are less likely to be picked again, so we propose a vector $\boldsymbol{q}_j$, where $\mathbb{E} = \left[h_1^e, h_2^e, \ldots, h_{|\mathcal{M}|}^e\right]^H$

$$\boldsymbol{q}_j = (\sum_{r=0}^{j} \boldsymbol{v}_r)^T \times \mathbb{E}$$

- Then $\boldsymbol{v}_{j+1}$ is calculated in an attentive manner with $\boldsymbol{q}_j$ as the attention query:

$$P(\boldsymbol{v}_{j+1,k} = 1 | \boldsymbol{v}_{1:j}) = \sigma(\mathbf{w}_v^T \boldsymbol{s}_j + \boldsymbol{q}_j \mathbf{W}^c \boldsymbol{h}_k^e)$$

Model

- **Content Planning: Context-Aware Keyphrase Selection.**
- utilize the binary cross-entropy loss with the gold-standard selection $\boldsymbol{v}_j^*$:

$$\mathcal{L}_{\text{sel}} = -\sum_{(\mathbf{x},\mathbf{y})\in D}\sum_{j=1}^{J}(\sum_{k=1}^{|\mathcal{M}|}\log(P(\boldsymbol{v}_{j,k}^*)))$$

Model

- **Style specification**

  predict a categorical style type $\widehat{\boldsymbol{t}_j}$ for each sentence:

$$\hat{\boldsymbol{t}}_j = \text{softmax}(\mathbf{w}_s^T(\tanh(\mathbf{W}^s[\boldsymbol{m}_j; \boldsymbol{s}_j]))$$

  where $\widehat{\boldsymbol{t}_j}$ is an estimated distribution over all types. The estimated $\widehat{\boldsymbol{t}_j}$ is compared against the gold-standard labels $\boldsymbol{t}_j^*$ to calculate a cross-entropy loss:

$$\mathcal{L}_{\text{style}} = -\sum_{(\mathbf{x},\mathbf{y})\in D}\sum_{j=1}^{J}\boldsymbol{t}_j^* \log \hat{\boldsymbol{t}}_j$$

Model

- **Style-Controlled Surface Realization**
- surface realization decoder is implemented with an LSTM, incorporating the content planning decoder hidden state $\boldsymbol{s}_{J(t)}$, J(t) is the sentence index, and and previously generated token y_{t-1}:

$$\boldsymbol{z}_t = g(\boldsymbol{z}_{t-1}, \tanh(\mathbf{W}^{ws}\boldsymbol{s}_{J(t)} + \mathbf{W}^{ww}\boldsymbol{y}_{t-1}))$$

For word prediction, we calculate two attentions, one over the input statement to get context vector $\boldsymbol{c}_t^w$, the other over the keyphrase memory bank $\mathcal{M}$, To better reflect the control over word choice by language styles, use style embedding $\boldsymbol{t}_{J(t)}$,

$$P(y_t|y_{1:t-1}) = \text{softmax}(\tanh(\mathbf{W}^o[\boldsymbol{z}_t; \boldsymbol{c}_t^w; \boldsymbol{c}_t^e; \boldsymbol{t}_{J(t)}])) \quad (9)$$

$$\boldsymbol{c}_t^w = \sum_{i=1}^{L} \alpha_i^w \boldsymbol{h}_i, \quad \alpha_i^w = \text{softmax}(\boldsymbol{z}_t \mathbf{W}^{wa} \boldsymbol{h}_i) \quad (10)$$

$$\boldsymbol{c}_t^e = \sum_{k=1}^{|\mathcal{M}|} \alpha_k \boldsymbol{h}_k^e, \quad \alpha_k^e = \text{softmax}(\boldsymbol{z}_t \mathbf{W}^{we} \boldsymbol{h}_k^e) \quad (11)$$

Model

- **Training Objective**

aggregating losses over (1) word generation:$L_{gen} = -\sum_D \sum_{t=1}^{T} \log(y_t^* | \boldsymbol{x}, \theta)$, (2) keyphrase selection: L_{sel} , (3) style prediction L_{style},

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{gen}}(\theta) + \gamma \cdot \mathcal{L}_{\text{style}}(\theta) + \eta \cdot \mathcal{L}_{\text{sel}}(\theta)$$

Datasets

Argument generation: Our first task is to generate a counter-argument for given statement on a controversial issue

(1) Input Keyphrases and Label Construction:For training, we construct a query per target argument sentence using its content words for retrieval (retrieve passages from Wikipedia, keep top 5 passages per query. For testing , use input statement for querying.

Then extract keyphrases from the retrieved passages based on topic signature words given input statement. a keyphrase is (a) a noun phrase or verb phrase that is shorter than 10 tokens (b) contains at least one content word; (c) has a topic signature or a Wikipedia title

A sentence is related to a keyphrase if there is any overlapping between the sentence and the keyphrase!

	Argument # Args (# Threads)	**Wikipedia** (Nor. / Sim.)	**AGENDA**
# Train	272,147 (11,434)	125,136	38,720
# Dev	40,291 (1,784)	21,004	1,000
# Test	46,757 (1,706)	23,534	1,000
# Tokens	54.87	70.57 / 48.60	141.34
# Sent.	2.48	3.15 / 3.20	5.59
# KP (candidates)	55.80	23.56	12.23
# KP (selected)	11.61	16.01/11.11	12.23

Table 1: Statistics of the three datasets. Average numbers are reported. For argument dataset, number of unique threads is also shown. On AGENDA, entities are extracted from abstract as keyphrases, hence all candidates are "selected".

Datasets

- **Argument generation**: Our first task is to generate a counter-argument for given statement on a controversial issue
- Sentence Style Label Construction:there are three sentence styles:
- (a) CLAIM is a proposition, usually containing one or two talking points, e.g., “I believe foreign aid is a useful bargaining chip”
- (b) P REMISE contains supporting arguments with reasoning or examples;
- (c) F UNCTIONAL is usually a generic statement, e.g., “I understand what you said”
- Rule based methods are used:

CLAIM: must be shorter than 20 tokens and matches any of the following patterns: (a) `i (don't)? (believe|agree|...)`; (b) `(anyone|all|everyone|nobody...) (should|could|need|must|might...)`;

PREMISE: must be longer than 5 tokens, contains at least one noun or verb content word, and matches any of the following patterns: (a) `(for (example|instance)|e.g.)`; (b) `(increase|reduce|improve|...)`

FUNCTIONAL: contains fewer than 5 alphabetical words and no noun or verb content word

Datasets

- **Paragraph Generation for Normal and Simple Wikipedia:** Input Keyphrases and Label Construction, Sentence Style Label Construction is similar to the first task
- **Paper Abstract Generation:** extract entities from articles as keyphrases

Results-Automatic Evaluation

- Consider a SEQ2SEQ as baseline , we implement a RETRIEVAL baseline, which returns the highest reranked passage retrieved with OP as the query.

	BLEU	ROUGE	MTR	Len.
RETRIEVAL	7.81	15.68	**10.59**	150.0
SEQ2SEQ	3.64	19.00	9.85	51.7
H&W (2018)	5.73	14.44	3.82	36.5
OURS (Oracle Plan.)	16.30*	20.25*	11.61	65.5
OURS	**13.19***	20.15*	10.42	65.2
w/o Style	12.61*	**20.28***	10.15	64.5
w/o Passage	11.84*	19.90*	9.03	62.6

Argument generation

	BLEU	ROUGE	METEOR	Length	BLEU	ROUGE	METEOR	Length
	Normal Wikipedia				Simple Wikipedia			
RETRIEVAL	20.10	28.60	12.23	44.5	21.99	33.44	12.97	34.7
SEQ2SEQ	22.62	27.49	14.74	52.9	21.98	29.36	16.94	52.8
LOGREGSEL	29.28	28.65	**27.76**	34.3	5.59	23.21	13.27	13.0
OURS (Oracle Plan.)	37.70*	45.41*	31.65*	79.8	34.22*	45.48*	32.84*	70.5
OURS	**33.76***	**40.08***	25.70	65.4	**31.22***	**40.76***	**26.76***	58.7
w/o Style	31.06*	37.72*	24.56	71.0	27.94*	38.20*	25.87*	64.5

Wikipedia generation

Result-Human Evaluation

- ask three proficient English speakers to assess the quality of generated arguments and Wikipedia paragraphs.
- Human subjects rate on a scale of 1 (worst) to 5 (best) on grammaticality, correctness and content richness.

	Argument			Wikipedia		
	Gram.	**Corr.**	**Cont.**	**Gram.**	**Corr.**	**Cont.**
HUMAN	4.81	3.90	3.48	4.84	4.73	4.49
OURS	**3.99***	**2.78***	**2.61***	**3.38**	**3.24***	3.43
w/o Style	3.03	2.26	2.03	2.99	2.89	**3.50**
Krippendorff's α	0.75	0.69	0.33	0.70	0.56	0.55

Table 6: Human evaluation on argument generation (Upper) and Wikipedia generation (Bottom). Grammaticality (**Gram**), correctness (**Corr**), and content richness (**Cont**) are rated on Likert scale ($1-5$). We mark our model with $*$ to indicate statistically significantly better ratings over the variant without style specification ($p < 0.001$, approximate randomization test).

Result-samples

Topic: Aborting a fetus has some non-zero negative moral implications
Human: It's not the birthing process that changes things. It's the existence of the baby. Before birth, the baby only exists inside another human being. After birth, it exists on its own in the world like every other person in the world.
Seq2seq: i 'm not going to try to change your view here , but i do n't want to change your position . i do n't think it 's fair to say that a fetus is not a person . it 's not a matter of consciousness .
Our model: tl ; dr : i agree with you , but i think it 's important to note that fetuses are not fully developed . i do n't know if this is the case , but it does n't seem to be a compelling argument to me at all , so i 'm not going to try to change your view by saying that it should be illegal to kill

Topic: Moon Jae-in
Simple Wikipedia: Moon Jae-in is a South Korean politician. He is the 12th and current President of South Korea since 10 May 2017 after winning the majority vote in the 2017 presidential election.
Seq2seq: moon election park is a election politician who served as prime minister of korea from 2007 to 2013 . he was elected as a member of the house of democratic party in the moon 's the the moon the first serving president of jae-in , in office since 2010 .
Our model: moon jae-in is a south korean politician and current president of south korea from 2012 to 2017 and again from 2014 to 2017.

Inspire

- Design a context planning decoder and jointly train the generation decoder and planning decoder.
- Use retrieval systems to label the keyphrases corresponding to each sentence

This week

- Tree decoder have no gold-standard labels for generated tree, try to obtain a content sequence to guide the construction of the tree
- "Pat's average driving speed is 25 km/hr faster than Kelly's, In the same time Pat drives 356 km and Kelly 256 km, what is Pat's driving speed“->

[('Pat', 'NNP'), ("'s", 'POS'), ('average', 'JJ'), ('driving', 'VBG'), ('speed', 'NN'), ('is', 'VBZ'), ('25', 'CD'), ('km/hr', 'NN'), ('faster', 'RBR'), ('than', 'IN'), ('Kelly', 'NNP'), ("'s", 'POS'), (',', ','), ('In', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('time', 'NN'), ('Pat', 'NNP'), ('drives', 'VBZ'), ('356', 'CD'), ('km', 'NN'), ('and', 'CC'), ('Kelly', 'NNP'), ('256', 'CD'), ('km', 'NN'), (',', ','), ('what', 'WP'), ('is', 'VBZ'), ('Pat', 'NNP'), ("'s", 'POS'), ('driving', 'VBG'), ('speed', 'NN')]

Label:[“var”,”operation”,”num”,”var”,”operation”,”var”,”var”,”operation”,”num”…]

This week

- Generate the context planning sequence by sequence labeling, but the accuracy of classifying can reach only about 50%
- (A) get high-quality labels
- (B) try generative method other than classifying