Learning a Simple and Effective Model for Multi-turn Response Generation with Auxiliary Tasks

Yufan Zhao[1], **Can Xu**[1*], **Wei Wu**[2], **Lei Yu**[3]
[1]Microsoft Corporation, Beijing, China
[2]Meituan, Beijing, China
[3]Beihang University, Beijing, China
{yufzhao,caxu}@microsoft.com
wuwei19850318@gmail.com
yulei@buaa.edu.cn

- We study multi-turn response generation for open-domain dialogues. The existing state-of-the-art addresses the problem with deep neural architectures. While these models improved response quality, their complexity also hinders the application of the models in real systems.
- In this work, we pursue a model that has a simple structure yet can effectively leverage conversation contexts for response generation. To this end, we propose four auxiliary tasks including word order recovery, utterance order recovery, masked word recovery, and masked utterance recovery

- The key idea is to transfer the burden of context understanding from modeling to learning by designing several auxiliary tasks, and leverage the auxiliary tasks as regularization in model estimation.
- Our contributions in the paper are three-fold: (1) proposal of balancing model complexity and model capability in multi-turn response generation; (2) proposal of four auxiliary learning tasks that transfer context understanding from modeling to learning; and (3) empirical verification of the effectiveness and the efficiency of the proposed model on three benchmarks.

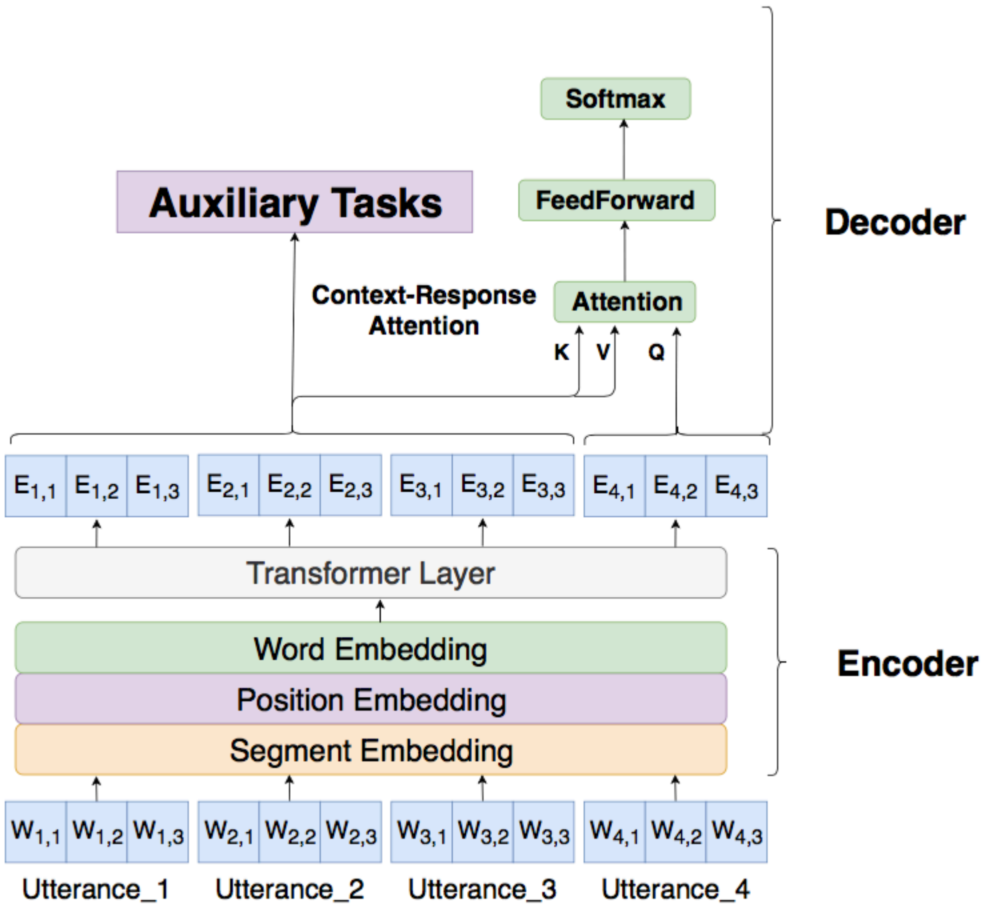

Figure 1: Architecture of the generation model.

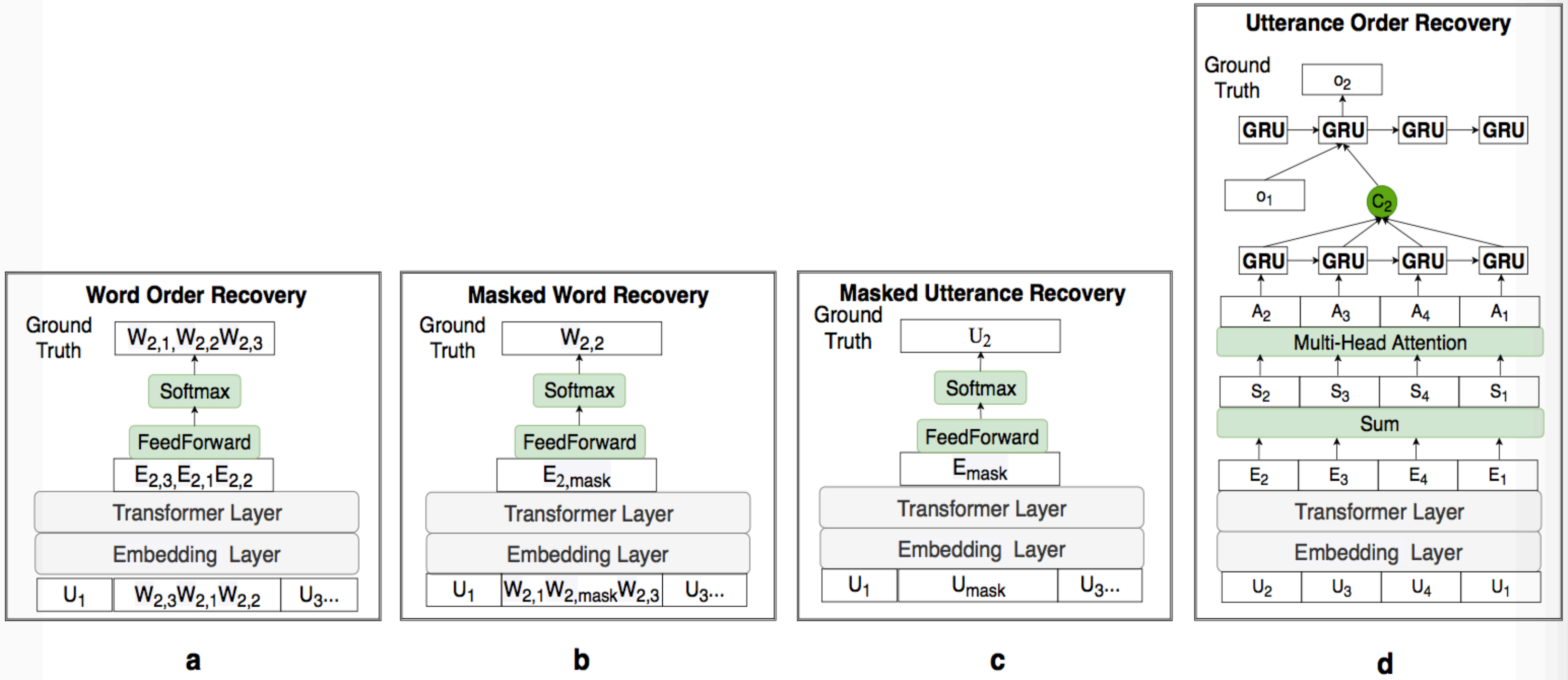

Figure 2: Auxiliary tasks.

Word order recovery

- 参数共用 和Decoder共用参数

- mask matrix

$$M_{ij} = \begin{cases} 0, & w_i \text{ and } w_j \text{ are in the same utterance,} \\ -\infty, & w_i \text{ and } w_j \text{ are in different utterances.} \end{cases}$$

Dataset	Model	PPL	BLEU	Distinct-1	Distinct-2	Average	Greedy	Extrema	Parameter size	Decoding speed
DailyDialog	HRED	56.22	0.535	1.553	3.569	81.393	65.546	48.109	34.5M	14.79ms
	HRAN	47.23	0.447	1.953	7.400	83.460	67.239	**49.599**	38.2M	17.15ms
	VHRED	44.79	0.997	1.299	6.113	83.866	67.186	48.570	34.8M	15.67ms
	SSN	44.28	1.250	2.309	7.266	72.796	**73.069**	44.260	20.0M	12.69ms
	ReCoSa	42.34	1.121	1.987	10.180	84.763	67.557	48.957	73.8M	40.89ms
	Our Model	**38.60**	**1.658**	**3.457**	**14.954**	**85.224**	69.518	49.069	20.3M/14.4M	12.15ms
PERSON-CHAT	HRED	46.04	1.279	0.164	0.450	83.329	64.486	47.132	28.3M	13.14ms
	HRAN	41.94	1.997	0.235	0.771	82.850	65.556	47.882	33.1M	18.43ms
	VHRED	42.07	2.181	0.312	1.915	82.995	65.578	46.810	28.8M	20.27ms
	SSN	47.90	2.288	0.637	2.623	**85.002**	66.752	47.461	15.2M	15.82ms
	ReCoSa	34.19	2.258	0.915	4.217	83.963	66.498	48.163	68.7M	39.38ms
	Our Model	**33.23**	**2.434**	**1.279**	**5.816**	83.632	**66.778**	**48.552**	18.4M/12.5M	13.89ms
Ubuntu	HRED	58.23	0.874	0.602	2.724	76.187	62.869	37.508	24.1M	25.09ms
	HRAN	48.14	0.922	0.472	2.217	76.654	62.145	37.282	29.5M	31.07ms
	VHRED	52.34	0.906	0.571	2.933	76.496	63.051	36.039	24.7M	30.47ms
	SSN	57.82	**1.681**	0.557	2.370	76.431	61.597	35.976	12.3M	21.11ms
	ReCoSa	43.67	0.911	0.722	4.439	77.619	**63.239**	36.742	60.6M	45.34ms
	Our Model	**40.94**	1.625	**0.783**	**5.151**	**78.754**	62.738	**38.538**	14.4M/8.5M	22.98ms

Table 2: Evaluation results on automatic metrics. Numbers in bold indicate the best performing model on the corresponding metrics.

- The Embedding Average (Average) metric projects the model response and ground truth response into two separate real-valued vectors by taking the mean over the word embeddings in each response, and then computes the cosine similarity between them.

- The Embedding Extrema (Extrema) metric similarly embeds the responses by taking the extremum (maximum of the absolute value) of each dimension, and afterwards computes the cosine similarity between them.

- The Embedding Greedy (Greedy) metric is more fine-grained; it uses cosine similarity between word embeddings to find the closest word in the human-generated response for each word in the model response.

DailyDialog				
models	win	loss	tie	kappa
Our Model v.s. HRED	0.42	0.13	0.45	0.675
Our Model v.s. VHRED	0.38	0.19	0.43	0.634
Our Model v.s. HRAN	0.31	0.16	0.53	0.587
Our Model v.s. SSN	0.36	0.22	0.42	0.638
Our Model v.s. ReCoSa	0.34	0.22	0.44	0.733
PERSONA-CHAT				
models	win	loss	tie	kappa
Our Model v.s. HRED	0.45	0.16	0.39	0.867
Our Model v.s. VHRED	0.39	0.21	0.40	0.650
Our Model v.s. HRAN	0.36	0.23	0.41	0.621
Our Model v.s. SSN	0.49	0.12	0.39	0.695
Our Model v.s. ReCoSa	0.39	0.29	0.32	0.566
Ubuntu				
models	win	loss	tie	kappa
Our Model v.s. HRED	0.49	0.14	0.37	0.692
Our Model v.s. VHRED	0.48	0.18	0.34	0.603
Our Model v.s. HRAN	0.47	0.13	0.40	0.612
Our Model v.s. SSN	0.45	0.18	0.37	0.698
Our Model v.s. ReCoSa	0.39	0.27	0.34	0.672

Table 4: Human evaluation results. The ratios are calculated by combining annotations from three judges together.

Discussions

- how do the simple architecture learned with the auxiliary tasks compare with a deep architecture ?
- if learning with the auxiliary tasks can also improve deep architectures

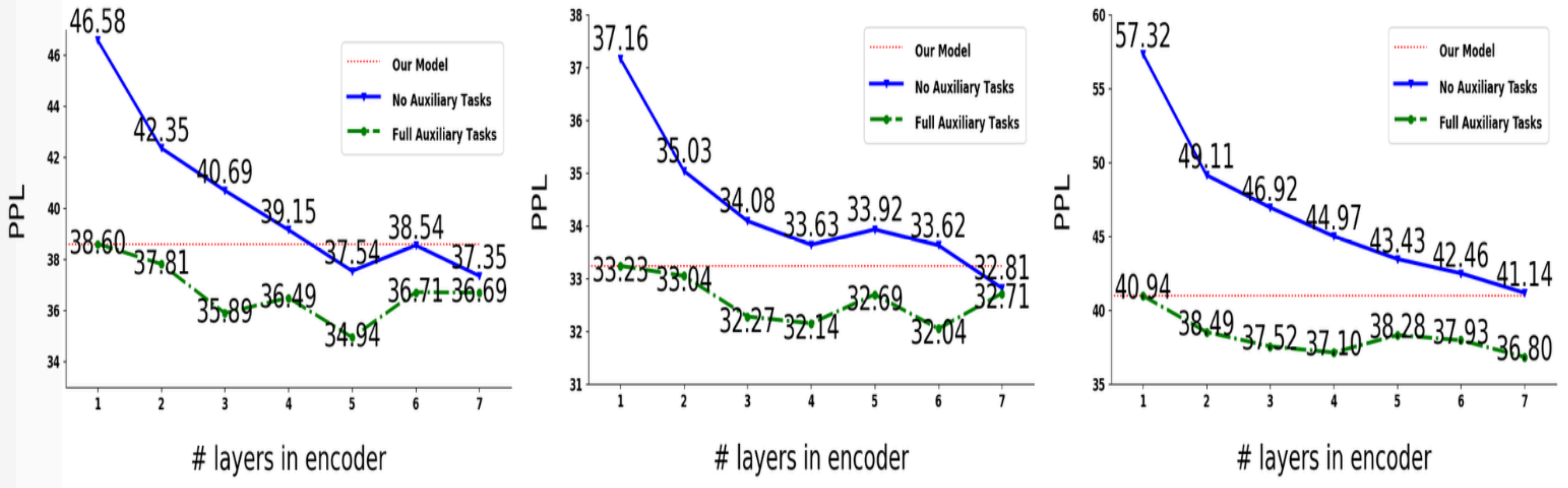

Figure 3: Performance of deep architectures. (a) DailyDialog; (b) PERSONA-CHAT; (c) Ubuntu

DailyDialog							
model variant	PPL	BLEU	distinct-1	distinct-2	Average	Greedy	Extrema
full tasks	38.60	1.658	3.457	14.954	85.224	69.518	49.069
- masked word recovery	38.37	1.365	2.629	11.135	85.270	69.901	49.495
- masked utterance recovery	39.06	1.407	2.980	12.544	85.143	69.667	49.791
- word order recovery	41.53	1.082	2.769	11.166	85.020	69.417	49.567
- utterance order recovery	38.69	1.215	2.551	9.764	85.253	69.678	49.644
- all tasks	46.58	0.903	1.775	7.136	84.042	69.017	48.467
PERSONA-CHAT							
model variant	PPL	BLEU	distinct-1	distinct-2	Average	Greedy	Extrema
full tasks	33.23	2.434	1.279	5.816	83.632	66.778	48.552
- masked word recovery	34.74	2.429	1.018	4.764	82.841	66.177	48.610
- masked utterance recovery	33.49	2.638	1.045	5.412	83.402	66.862	48.810
- word order recovery	35.06	2.355	1.028	4.698	82.503	66.011	48.350
- utterance order recovery	33.24	2.484	1.054	5.011	82.652	66.025	47.927
- all tasks	37.16	1.928	0.938	4.141	82.104	65.899	47.162
Ubuntu							
model variant	PPL	BLEU	distinct-1	distinct-2	Average	Greedy	Extrema
full tasks	40.94	1.625	0.783	5.151	78.754	62.738	38.538
- masked word recovery	47.02	1.135	0.404	2.195	74.735	61.683	37.914
- masked utterance recovery	42.48	1.543	0.519	2.419	76.381	62.203	37.482
- word order recovery	48.57	0.962	0.325	1.537	77.615	62.819	38.651
- utterance order recovery	52.04	1.023	0.359	1.609	74.982	59.384	36.825
- all tasks	57.32	0.851	0.391	1.765	73.582	62.581	37.268

Table 3: Results of ablation study.

Acrostic Poem Generation

Rajat Agarwal
New York University
rajat.agarwal@nyu.edu

Katharina Kann
University of Colorado Boulder
katharina.kann@colorado.edu

Poetry around the city
Opera the poet sings
Essay on man epistle by
Translated kings.

Figure 1: An acrostic poem generated by our proposed baseline model for the word *poet*.

- A conditional neural language model, which generates an acrostic poem based on a given word.
- A rhyming model, trained on sonnets, which gener- ates rhyming words for the last position in each line.
- we feed the word embedding of the topic to the language model at each time step.

Number of lines	**4**	**5**	**6**	**7**	**8**	**Total**
KnownTopicPoems	30,433	5,413	7,233	4,795	6,098	53,972
UnknownTopicPoems	26,986	10,765	11,609	6,433	9,487	65,280
Total	57,419	16,178	18,842	11,228	15,585	**119,252**

Table 1: Number of poems in our datasets used for training, listed by the number of lines they contain.

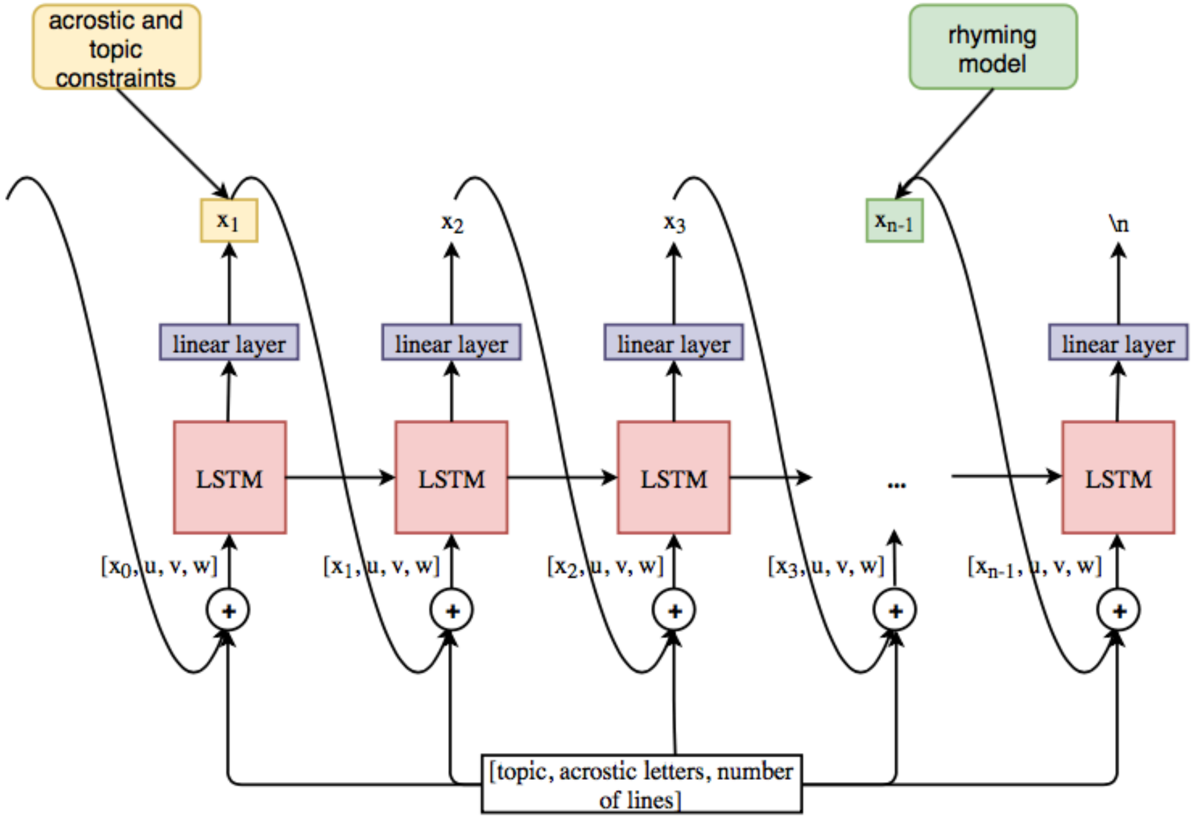

Figure 2: Overview of the baseline model we introduce together with the task of acrostic poem generation.

Neural Poem Language Model

$$p(x) = \prod_{i=1}^{n} p(x_i | \{x_0, ...x_{i-1}\}, u, v, w) \qquad (1)$$

U is a given topic,
V is the acrostic word,
W is the number of lines

For generation – but not during training –, u and v correspond to the same word.

Each letter of the acrostic word v is represented as a one-hot vector of size 27

The number of lines is represented in the model by a single-digit tensor.

The first word of each line.

First, from all words in our vocabulary which start with the indicated character, we compute the $k = 5$ nearest neighbors $n_1, \ldots, n_k$ to the topic word u, using cosine similarity and our pretrained embeddings:

$$\mathrm{sim}(x, u) = \frac{\mathrm{emb}(x) \cdot \mathrm{emb}(u)}{\|\mathrm{emb}(x)\| \cdot \|\mathrm{emb}(u)\|}$$

Then, we select our output with a probability of $m_1 = 0.7$ as

$$\mathrm{argmax}_i (p_{LM}(n_1), p_{LM}(n_2), \ldots, p_{LM}(n_k)) \tag{3}$$

However, this can cause the output to frequently become incoherent. Thus, we sample the first word from the language model, masking out all words that start with a wrong letter, with a probability of $m_2 = 0.3$.[6]

Rhyming Model

- 4 lines: ABAB; 5 lines: ABABC; 6 lines: ABABCC; 7 lines: ABABCDC; 8 lines: ABABCDCD.
- 在对应处结尾生成

Whenever a rhyming word is required, our rhyming model computes the probability of an output word c, consisting of a character sequence $c_1 c_2 ... c_l$, as:

$$p(c) = \prod_{i=1}^{l} p(c_i | \{c_0, ...c_{i-1}\}, a, b) \tag{4}$$

Model	**Perplexity**
GOLD+	24.22
GOLD-	23.79
PRED/GOLD+	19.94
PRED/GOLD-	18.79
WIKI+	**16.87**
WIKI-	18.19

Table 2: Perplexity on the test set of KnownTopicPoems for all language models; best score in bold.

	All				Known♡				Unknown♠			
	F	**M**	**P**	**A**	**F**	**M**	**P**	**A**	**F**	**M**	**P**	**A**
Human	4.1	3.95	4.22	3.67	4.1	3.95	4.22	3.67	-	-	-	-
NeuralPoet	3.48	2.75	**3.66**	2.55	**3.70**	2.86	**3.77**	2.79	3.25	2.63	3.56	2.31
NeuralPoet-ST	3.51	2.79	3.25	2.59	3.39	2.81	3.31	2.73	**3.62**	2.76	3.20	2.43
NeuralPoet-ST-AC	**3.60**	2.95	3.59	2.62	3.58	**3.12**	3.35	2.70	**3.62**	3.03	**3.83**	2.56
NeuralPoet-ST-RH	3.36	2.94	3.32	2.54	3.40	2.99	3.41	2.69	3.32	2.89	3.27	2.38
NeuralPoet-ST-TP	**3.60**	**3.11**	3.52	**2.87**	**3.70**	3.06	3.57	**2.84**	3.50	**3.15**	3.48	**2.90**

Table 3: Human evaluation and ablation study; *F* = Fluency; *M* = Meaning; *P* = Poeticness; *A* = Overall; *ST*=selecting first words for each line according to the acrostic; *AC*=acrostic forcing; *RH*=rhyming model; *TP*=feeding of topic vector.

alone♡

Alone we spoke,
Less, do not fear my heart,
Only later, i may not love,
Not to have hoped that i would not apart,
Even... i am sure.

nature♡

Not still a child
Am i one of you
That look in the wild
Upon your paradise full of view
Remember my soul 's face well
Experience 's as shall.

cake♠

Chocolate wall and marble cup
Apples howl with golden hair
Kitchen of the world they stir