Search-Based Unsupervised Text Generation

Lili Mou

Dept. Computing Science, University of Alberta
Alberta Machine Intelligence Institute (Amii)

doublepower.mou@gmail.com

Outline

- Introduction
- General framework
- Applications
  - Paraphrasing
  - Summarization
  - Text simplification
- Conclusion & Future Work

A fading memory …

- Of how I learned natural language processing (NLP):

$$\text{NLP} = \underbrace{\text{NLU}}_{\textbf{Understanding}} + \underbrace{\text{NLG}}_{\textbf{Generation}}$$

  - NLU was the main focus of NLP research.
  - NLG was relatively easy, as we can generate sentences by rules, templates, etc.

- Why this may NOT be correct?
  - Rules and templates are not natural language.
  - How can we represent meaning? — Almost the same question as NLU.

A fading memory …

- Of how I learned natural language processing (NLP):

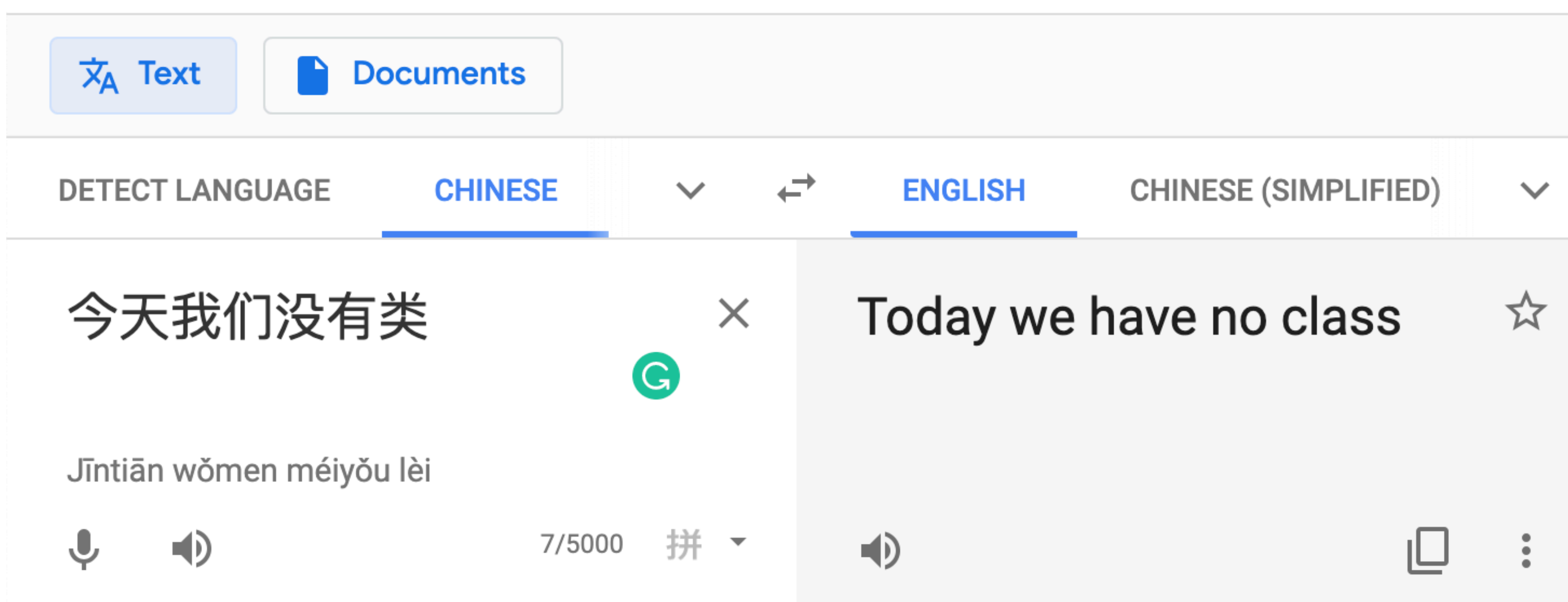
$$\text{NLP} = \underbrace{\text{NLU}}_{\textbf{Understanding}} + \underbrace{\text{NLG}}_{\textbf{Generation}}$$

  - NLU was the main focus of NLP research.
  - NLG was relatively easy, as we can generate sentences by rules, templates, etc.

- Why this may NOT be correct?
  - Rules and templates are not natural language.
  - How can we represent meaning? — Almost the same question as NLU.

Why NLG is interesting?

- Industrial applications
  - Machine translation
  - Headline generation for news
  - Grammarly: grammatical error correction



https://translate.google.com/

Why NLG is interesting?

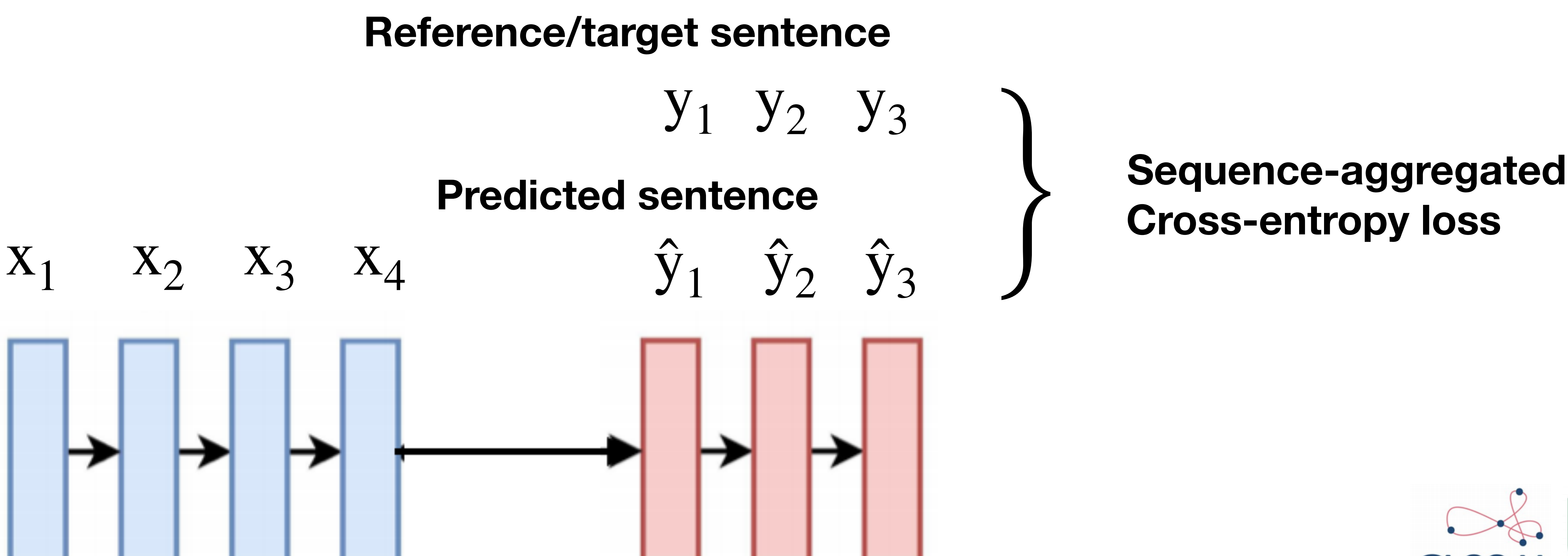
- Industrial applications
  - Machine translation
  - Headline generation for news
  - Grammarly: grammatical error correction
- Scientific questions
  - Non-linear dynamics for long-text generation
  - Discrete “multi-modal” distribution

Supervised Text Generation

Sequence-to-sequence training

Training data = $\{(\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}_{m=1}^{M}$

known as a *parallel corpus*



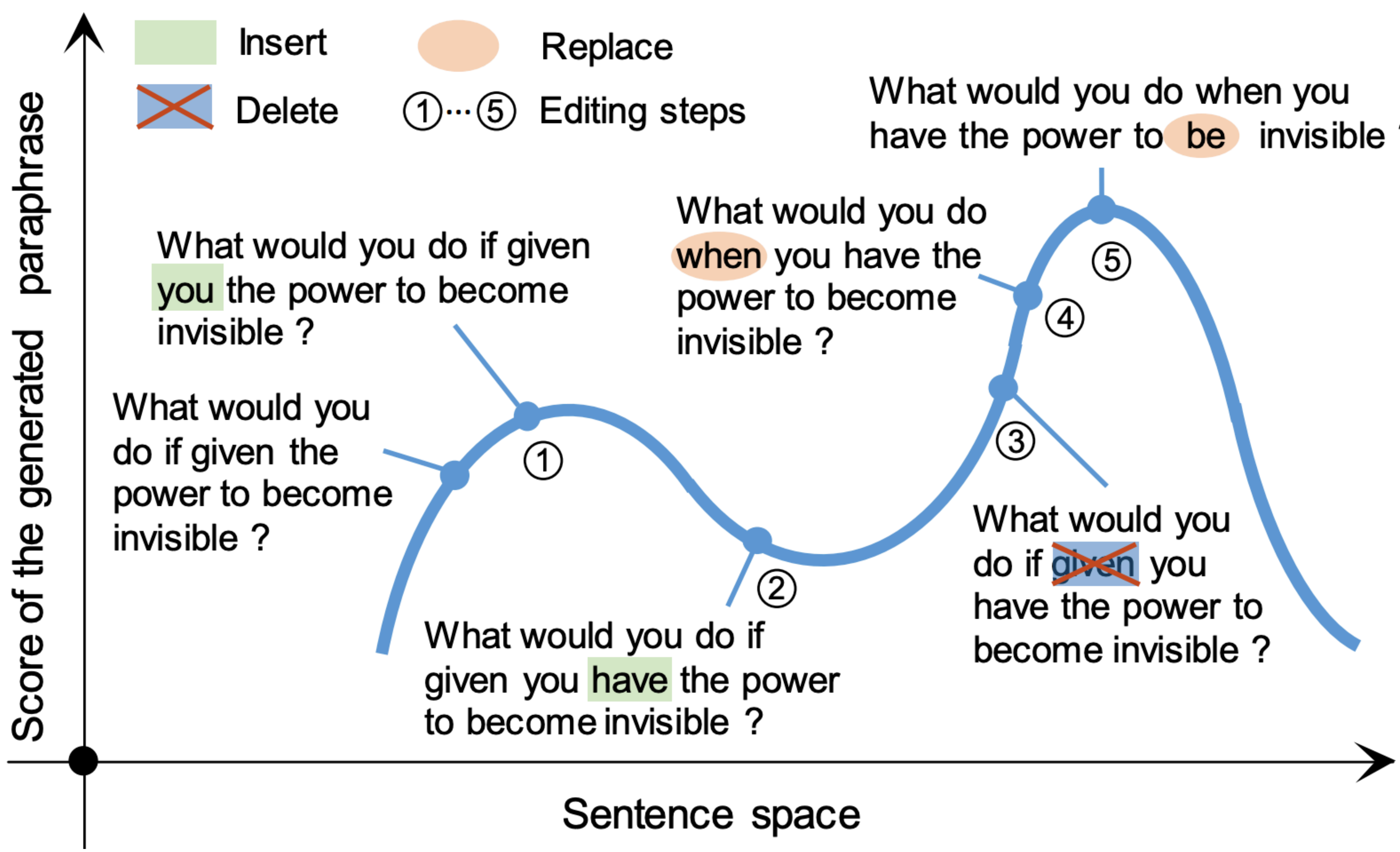
Unsupervised Text Generation

- Training data = $\{\mathbf{x}^{(m)}\}_{m=1}^{M}$
  - Not even training (we did it by searching)
- Important to **industrial applications**
  - Startup: No data
  - Minimum viable product
- Scientific interest
  - How can AI agents go beyond NLU to NLG?
  - Unique search problems

General Framework

General Framework

- **Search objective**
  - Scoring function measuring text quality
- **Search algorithm**
  - Currently we are using stochastic local search



Scoring Function

- Search objective
  - Scoring function measuring text quality

$$s(\mathbf{y}) = s_{LM}(\mathbf{y}) \cdot s_{Semantic}(\mathbf{y})^{\alpha} \cdot s_{Task}(\boldsymbol{y})^{\beta}$$

- Language fluency

- Semantic coherence

- Task-specific constraints

Scoring Function

- Search objective
  - Scoring function measuring text quality

$$s(\mathbf{y}) = s_{LM}(\mathbf{y}) \cdot s_{Semantic}(\mathbf{y})^{\alpha} \cdot s_{Task}(\mathbf{y})^{\beta}$$

- **Language fluency**
  - Language model estimates the “probability" of a sentence

$$\overleftrightarrow{\mathrm{PPL}}(\mathbf{y}) = \sqrt[2|\mathbf{y}|]{\prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overrightarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{<i})} \prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overleftarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{>i})}}.$$

$$s_{LM}(\mathbf{y}) = \mathrm{PPL}(\mathbf{y})^{-1}$$

- Semantic coherence

- Task-specific constraints

Scoring Function

- Search objective
  - Scoring function measuring text quality

$$s(\mathbf{y}) = s_{LM}(\mathbf{y}) \cdot s_{Semantic}(\mathbf{y})^{\alpha} \cdot s_{Task}(\boldsymbol{y})^{\beta}$$

- Language fluency

- **Semantic coherence**

$$s_{semantic} = \cos(\boldsymbol{e}(\mathbf{y}), \boldsymbol{e}(\mathbf{y}))$$

- Task-specific constraints

Scoring Function

- Search objective
  - Scoring function measuring text quality

$$s(\mathbf{y}) = s_{LM}(\mathbf{y}) \cdot s_{Semantic}(\mathbf{y})^{\alpha} \cdot s_{Task}(\boldsymbol{y})^{\beta}$$

- Language fluency

- Semantic coherence

- Task-specific constraints
  - Paraphrasing: lexical dissimilarity with input
  - Summarization: length budget

Search Algorithm

- Observations:
  - The output closely resembles the input
  - Edits are mostly local
  - May have hard constraints

- Thus, we mainly used **local stochastic search**

Search Algorithm

(stochastic local search)

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within budget at step t:

$\mathbf{y}' \sim \mathrm{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Search Algorithm

Local edits for $\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$

- General edits
  - Word deletion
  - Word insertion
  - Word replacement

$$p(w_*|\cdot) = \frac{f_{\text{sim}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{exp}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{flu}}(\mathrm{x}_*)}{Z},$$

$$Z = \sum_{w_* \in \mathcal{W}} f_{\text{sim}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{exp}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{flu}}(\mathrm{x}_*),$$

Gibbs in Metropolis

- Task specific edits
  - Reordering, swap of word selection, etc.

Search Algorithm

Example: Metropolis—Hastings sampling

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within your budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$

$$\boxed{\begin{aligned} A(\mathrm{x}'|\mathrm{x}_{t-1}) &= \min\{1, A^*(\mathrm{x}'|\mathrm{x}_{t-1})\} \\ A^*(\mathrm{x}'|\mathrm{x}_{t-1}) &= \frac{\pi(\mathrm{x}')g(\mathrm{x}_{\mathrm{t}-1}|\mathrm{x}')}{\pi(\mathrm{x}_{t-1})g(\mathrm{x}'|\mathrm{x}_{t-1})} \end{aligned}}$$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Search Algorithm

Example: Simulated annealing

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within your budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$ $\boxed{p(\text{accept}|\mathrm{x}_*, \mathrm{x}_t, T) = \min\left(1, e^{\frac{f(\mathrm{x}_*) - f(\mathrm{x}_t)}{T}}\right)}$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Search Algorithm

Example: Hill climbing

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within your budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$

whenever $\mathbf{y}'$ is better than $\mathbf{y}_{t-1}$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Applications

Paraphrase Generation

Input	Reference
Which is the best training institute in Pune for digital marketing ?	Which is the best digital marketing training institute in Pune ?

Could be useful for various NLP applications

- E.g., query expansion, data augmentation

Paraphrase Generation

- Search objective
  - Fluency
  - Semantic preservation
  - Expression diversity
    - The paraphrase should be different from the input

$$s_{exp}(\mathbf{y}_*, \mathbf{y}_0) = 1 - \mathrm{BLEU}(\mathbf{y}_*, \mathbf{y}_0)$$

BLEU here measures the *n*-gram overlapping

- Search algorithm
- Search space
- Search neighbors

Paraphrase Generation

- Search objective
  - Fluency
  - Semantic preservation
  - Expression diversity
    - The paraphrase should be different from the input

$$s_{exp}(\mathbf{y}_*, \mathbf{y}_0) = 1 - \mathrm{BLEU}(\mathbf{y}_*, \mathbf{y}_0)$$

BLEU here measures the *n*-gram overlapping

- Search algorithm: Simulated annealing
- Search space: the entire sentence space with $\mathbf{y}_0$ = input
- Search neighbors
  - Generic word deletion, insertion, and replacement
  - Copying words in the input sentence

Text Simplification

Input	Reference
In 2016 alone, American developers had spent 12 billion dollars on **constructing** theme parks, according to a Seattle based reporter.	American developers had spent 12 billion dollars in 2016 alone on **building** theme parks.

Could be useful for

- education purposes (e.g., kids, foreigners)
- for those with dyslexia

Key observations

- Dropping phrases and clauses
- Phrase re-ordering
- Dictionary-guided lexicon substitution

Text Summarization

Search objective

- Language model fluency (discounted by word frequency)
- Cosine similarity
- Entity matching
- Length penalty
- Flesh Reading Ease (FRE) score [Kincaid et al., 1975]

Search operations

Text Summarization

Search objective

- Language model fluency (discounted by word frequency)
- Cosine similarity
- Entity matching
- Length penalty
- Flesh Reading Ease (FRE) score [Kincaid et al., 1975]

Search operations

- Dictionary-guided substitution (e.g., WordNet)
- Phrase removal
- Re-ordering

with parse trees

Text Summarization

Input	Reference
The *world's biggest miner* ***bhp billiton*** *announced tuesday it was* ***dropping*** *its controversial hostile* ***takeover bid*** *for rival* ***rio tinto*** *due to the state of the global economy*	bhp billiton drops rio tinto takeover bid

Key observation

- Words in the summary mostly come from the input
- If we generate the summary by selecting words, we have

```
bhp billiton dropping hostile bid for rio tinto
```

Text Summarization

- Search objective
  - Fluency
  - Semantic preservation
  - A hard length constraint

$$f_{\text{LEN}}(\mathbf{y}; s) = \begin{cases} 1, & \text{if } |\mathbf{y}| = s, \\ -\infty, & \text{otherwise.} \end{cases}$$

  (Explicitly controlling length is not feasible in previous work)

- Search space
- Search neighbor
- Search algorithm

Text Summarization

- Search objective
  - Fluency
  - Semantic preservation
  - A hard length constraint

$$f_{\text{LEN}}(\mathbf{y}; s) = \begin{cases} 1, & \text{if } |\mathbf{y}| = s, \\ -\infty, & \text{otherwise.} \end{cases}$$

  (Explicitly controlling length is not feasible in previous work)

- Search space with only feasible solutions

$$|\mathscr{V}|^{|\mathbf{y}|} \Longrightarrow \binom{|\mathbf{x}|}{s}$$

- Search neighbor: swap only

- Search algorithm: hill-climbing

Experimental Results

Research Questions

- General performance
- Greediness vs. Stochasticity
- Search objective vs. Measure of success

General Performance

Paraphrase generation

		Quora				Wikianswers			
	Model	iBLEU	BLEU	Rouge1	Rouge2	iBLEU	BLEU	Rouge1	Rouge2
Supervised	ResidualLSTM	12.67	17.57	59.22	32.40	22.94	27.36	48.52	18.71
	VAE-SVG-eq	15.17	20.04	59.98	33.30	26.35	32.98	50.93	19.11
	Pointer-generator	16.79	22.65	61.96	36.07	31.98	39.36	57.19	25.38
	Transformer	16.25	21.73	60.25	33.45	27.70	33.01	51.85	20.70
	Transformer+Copy	17.98	24.77	63.34	37.31	31.43	37.88	55.88	23.37
	DNPG	**18.01**	**25.03**	**63.73**	**37.75**	**34.15**	**41.64**	**57.32**	**25.88**
Supervised + Domain-adapted	Pointer-generator	5.04	6.96	41.89	12.77	21.87	27.94	53.99	20.85
	Transformer+Copy	6.17	8.15	44.89	14.79	23.25	29.22	53.33	21.02
	Shallow fusion	6.04	7.95	44.87	14.79	22.57	29.76	53.54	20.68
	MTL	4.90	6.37	37.64	11.83	18.34	23.65	48.19	17.53
	MTL+Copy	7.22	9.83	47.08	19.03	21.87	30.78	54.10	21.08
	DNPG	10.39	16.98	56.01	28.61	25.60	35.12	56.17	23.65
Unsupervised	VAE	8.16	13.96	44.55	22.64	17.92	24.13	31.87	12.08
	Lag VAE	8.73	15.52	49.20	26.07	18.38	25.08	35.65	13.21
	CGMH	9.94	15.73	48.73	26.12	20.05	26.45	43.31	16.53
	UPSA	12.03	18.21	59.51	32.63	24.84	32.39	54.12	21.45

BLEU and ROUGE scores are automatic evaluation metrics based on references

amii

General Performance

Text Summarization

	Model	Data			Len D	Rouge F1			Len O
		article	title	external		R-1	R-2	R-L	
	Lead-N-8	✓			8	21.39	7.42	20.03	7.9
A	*HC_article_8*	✓			8	23.09	7.50	21.29	7.9
	HC_title_8		✓		8	**26.32**	**9.63**	**24.19**	7.9
	Lead-N-10	✓			10	23.03	7.95	21.29	9.8
	Wang and Lee (2018)	✓	✓		-	27.29	10.01	24.59	10.8
	Zhou and Rush (2019)		✓	billion	-	26.48	10.05	24.41	9.3
B	*HC_article_10*	✓			10	24.44	8.01	22.21	9.8
	HC_title_10		✓		10	27.52	10.27	24.91	9.8
	HC_title+twitter_10		✓	twitter	10	28.26	10.42	25.43	9.8
	HC_title+billion_10		✓	billion	10	**28.80**	**10.66**	**25.82**	9.8
	Lead-P-50	✓			50%	24.97	8.65	22.43	14.6
	Fevry and Phang (2018)	✓		SNLI	50%	23.16	5.93	20.11	14.8
C	Baziotis et al. (2019)	✓			50%	24.70	7.97	22.14	15.1
	HC_article_50p	✓			50%	25.58	8.44	22.66	14.9
	HC_title_50p		✓		50%	**27.05**	**9.75**	**23.89**	14.9

General Performance

Text Simplification

Method	BLEU	SARI	Add	Delete	Keep	GM	FKGL	Len
Reference	100	70.13	-	-	-	83.74	3.20	12.75
Baselines								
Complex	21.30	2.82	-	-	-	7.75	8.62	23.06
Reduced-250	11.79	28.39	-	-	-	18.29	-0.23	14.48
Supervised Methods								
PBMT-R	18.1	15.77	3.07	38.34	5.90	16.89	7.59	23.06
Hybrid	14.46	28.61*	0.95*	78.86*	6.01*	20.34	4.03	12.41
EncDecA	21.68	24.12	2.73	62.66	6.98	22.87	5.11	16.96
Dress	23.2	27.37	**3.08**	71.61	7.43	25.2	4.11	14.2
Dress-Ls	24.25	26.63	3.21	69.28	7.4	25.41	4.21	14.37
DMass	11.92	31.06	1.25	84.12	7.82	19.24	3.60	15.07
S2S-All-FA	19.55	30.73	2.64	81.6	**7.97**	24.51	2.60	10.81
Edit-NTS	19.85	30.27*	2.71*	80.34*	7.76*	24.51	3.41	10.92
EncDecP	23.72	28.31	-	-	-	25.91	-	-
EntPar	11.14	**33.22**	2.42	**89.32**	7.92	19.24	1.34	7.88
Unsupervised Methods (Ours)								
Base	**27.22**	26.07	2.35	68.35	7.5	26.64	2.95	12.9
Base+LS	27.17	26.26	2.28	68.94	7.57	**26.71**	2.93	12.88
Base+RO	26.31	26.99	2.47	70.88	7.63	26.64	3.14	12.81
Base+LS+RO	26.21	27.11	2.40	71.26	7.67	26.66	3.12	12.81

General Performance

Human evaluation on paraphrase generation

Model	Relevance		Fluency	
	Mean Score	Agreement	Mean Score	Agreement
VAE	2.65	0.41	3.23	0.51
Lag VAE	2.81	0.45	3.25	0.48
CGMH	3.08	0.36	3.51	0.49
UPSA	**3.78**	0.55	**3.66**	0.53

General Performance

Examples

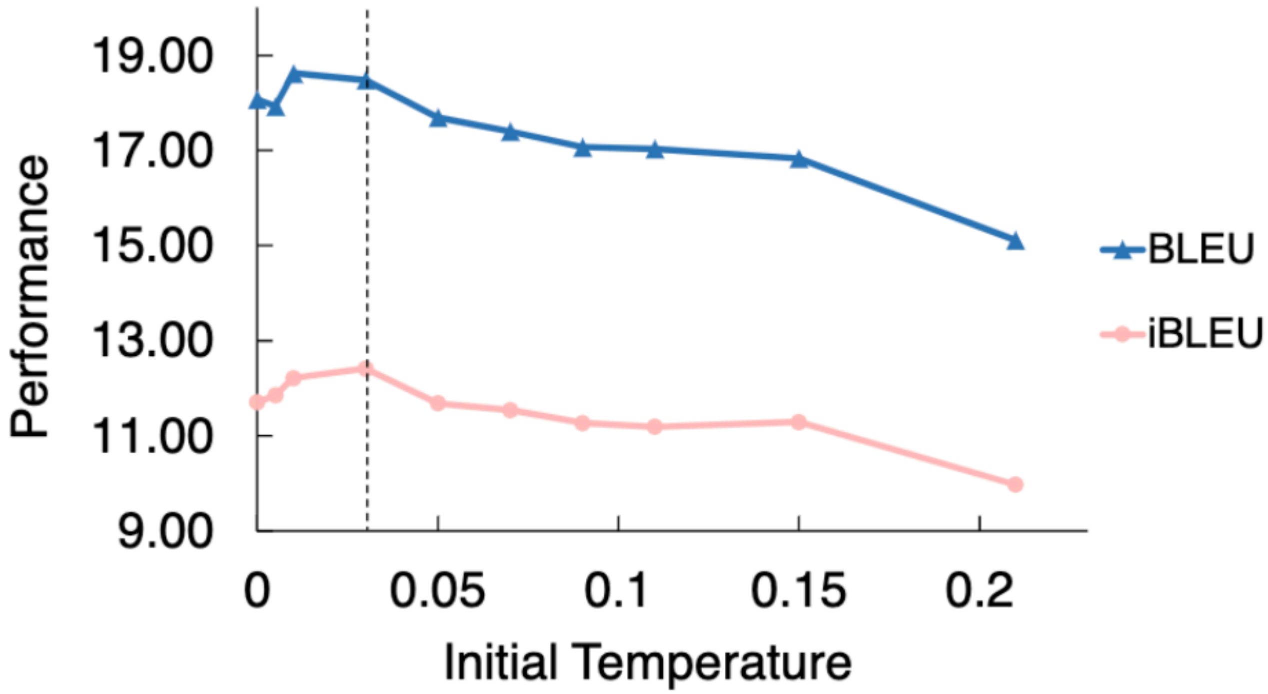
Input	VAE	Lag VAE	CGMH	UPSA
where are best places for spring snowboard-ing in the us?	where are best places for running in the world? (3.33)	where are best places for honeymoon year near the us? (2.33)	Where is best store for the snowboarding in the US? (3.67)	Where can I find the best places in the US for snow-boarding? (4.67)
how can i become good in studies?	how can i have a good an-droid phone? (2.33)	how can i become good students? (4.33)	how can i become very rich in studies? (4.00)	how should i do to get better grades in my studies? (4.33)
what are the pluses and minuses about life as a foreigner in singapore?	what are the UNK and most interesting life as a foreigner in medieval greece? (2.33)	what are the UNK and in-teresting things about life as a foreigner? (2.33)	what are the misconception about UNK with life as a foreigner in western? (2.33)	what are the mistakes and pluses life as a foreigner in singapore? (2.67)

Main conclusion

- Search-based unsupervised text generation works in a variety of applications
- Surprisingly, it does yield **fluent sentences**.

Greediness vs Stochasticity

Paraphrase generation



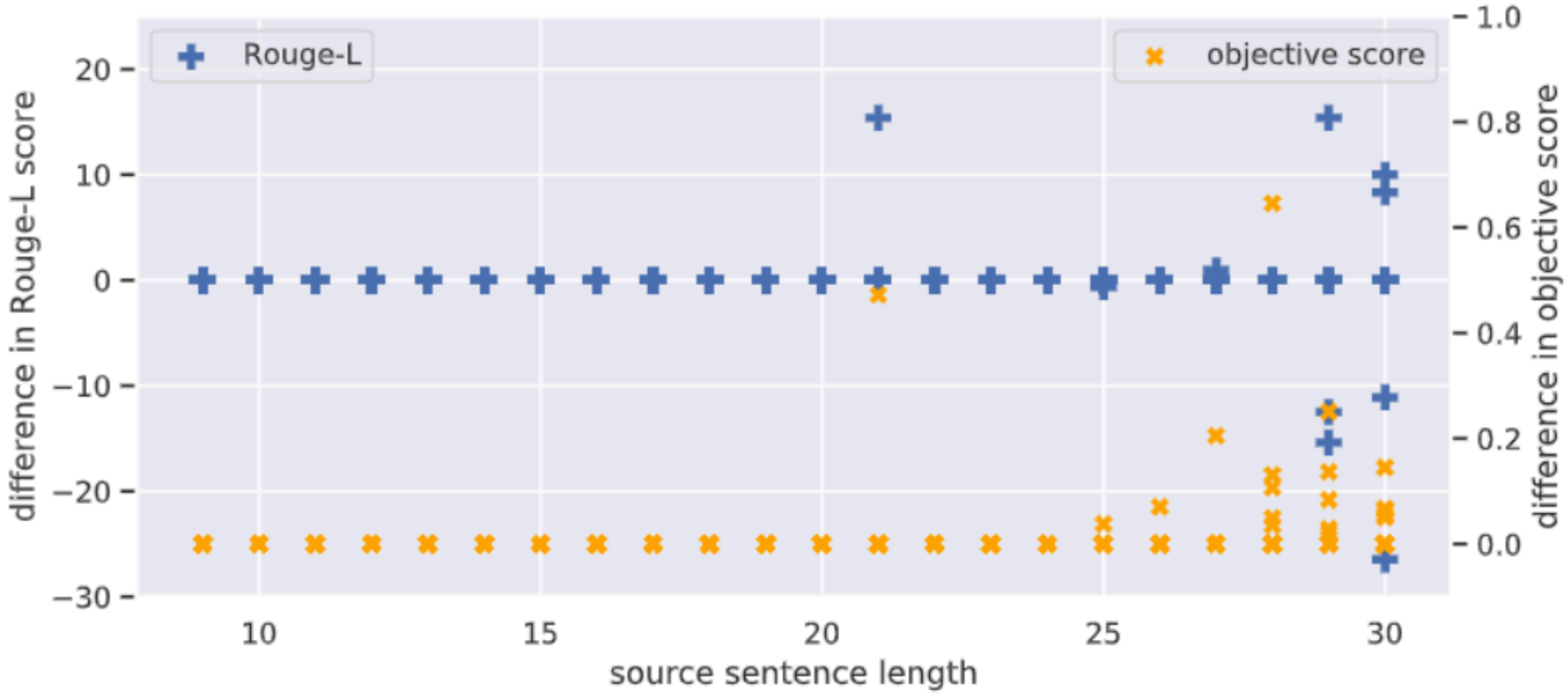
Line #	UPSA Variant	iBLEU	BLEU	Rouge1	Rouge2
1	UPSA	**12.41**	18.48	57.06	31.39
2	w/o $f_{\text{sim,key}}$	10.28	15.34	50.85	26.42
3	w/o $f_{\text{sim,sen}}$	11.78	17.95	57.04	30.80
4	w/o f_{exp}	11.93	21.17	59.75	34.91
5	w/o copy	11.42	17.25	56.09	29.73
6	w/o annealing	10.56	16.52	56.02	29.25

Findings:

- Greedy search $\prec$ Simulated annealing
- Sampling $\prec$ stochastic search

Search Objective vs. Measure of Success

Experiment: summarization by word selection



Comparing hill-climbing (w/ restart) and exhaustive search

- Exhaustive search does yield higher scores $s(\mathbf{y})$
- Exhaustive search does NOT yield higher measure of success (ROUGE)

Conclusion & Future Work

Search-based unsupervised text generation

General framework

- Search objective
  - fluency, semantic coherence, etc.
- Search space
  - word generation from the vocabulary, word selection
- Search algorithm
  - Local search with word-based edit
  - MH, SA, and hill climbing

Applications

- Paraphrasing, summarization, simplification

Future Work

<u>Defining the search neighborhood</u>

Input: What would you do **if given the** power to become invisible?

Output: What would you **do when you have** the power to be invisible?

Current progress

- Large edits are possibly due to the less greedy SA but are rare

Future work

- Phrase-based edit (combining discrete sampling with VAE)
- Syntax-based edit (making use of probabilistic CFG)

Future Work

<u>Initial state of the local search</u>

Current applications

- Paraphrasing, summarization, text simplification, grammatical error correction
- Input and desired output closely resemble each other

Future work

- Dialogue systems, machine translation, etc.
- Designing initial search state for general-purpose TextGen
- Combining retrieval-based methods

Future Work

Combining search and learning

Disadvantage of search-only approaches

- Efficiency: 1—2 seconds per sample
- Heuristically defined objective may be deterministically wrong

Future work

- MCTS (currently exploring)
- Difficulties: large branching factor, noisy reward

References

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, Lei Li. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. In *AAAI*, 2019.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou and Sen Song. Unsupervised paraphrasing by simulated annealing. In *ACL*, 2020.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova and Katja Markert. Discrete optimization for unsupervised sentence summarization with word level extraction. In ACL, 2020.

Dhruv Kumar, Lili Mou, Lukasz Golab and Olga Vechtomova. Iterative edit-based unsupervised sentence simplification. In *ACL*, 2020.

Acknowledgments

Lili Mou is supported by AltaML, Amii Fellow Program, and Canadian CIFAR AI Chair Program.

Q&A

Thanks for listening!