自然語言處理與應用
Natural Language Processing and Applications

NLG Evaluations

Instructor: 林英嘉 (Ying-Jia Lin)
2025/04/21

Course GitHub

Slido # NLP_0421

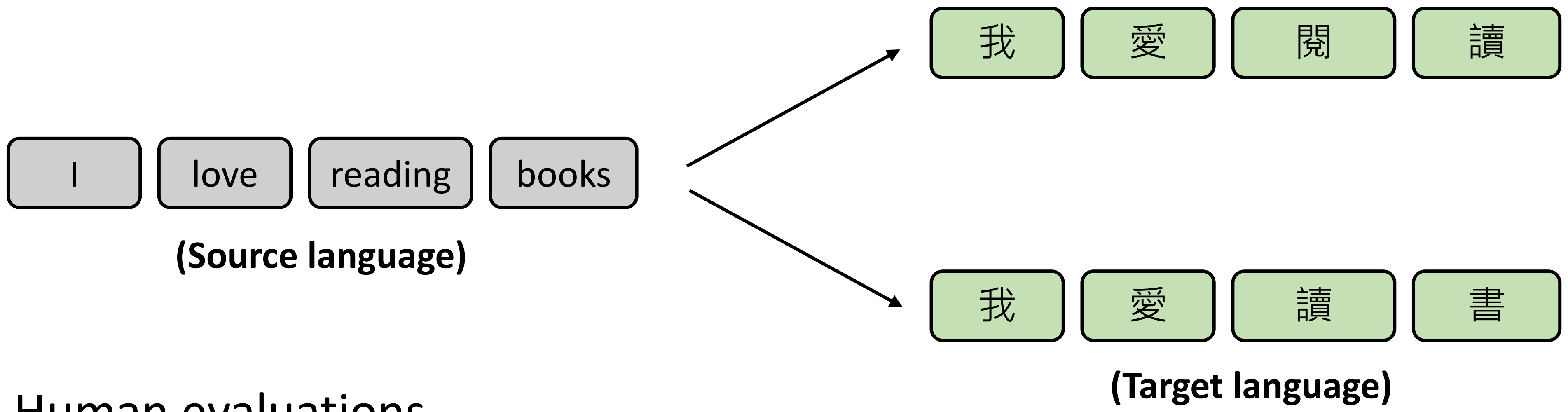
Evaluations

- Perplexity
- BLEU Score
- ROUGE Score
- BERTScore
- BLEURT

How to evaluate natural language generation?

- Natural language is hard to evaluate due to <u>subjectivity</u> and language <u>diversity</u>.

For example: Machine Translation

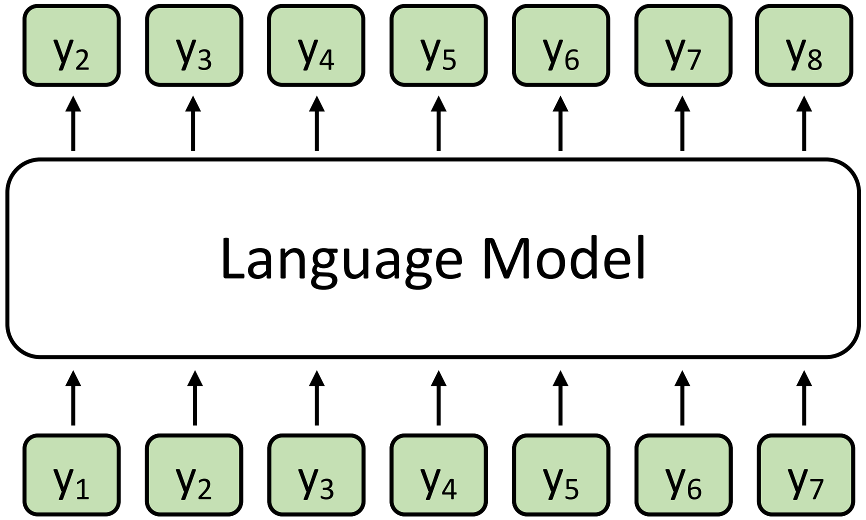


- Human evaluations
- Automatic evaluations (We will focus on this topic.)

[Recap] Language Modeling

$P(y_t|y_1, y_2, \dots, y_{t-1})$ ← Next-token prediction

$P(y_2|\ y_1)$ $P(y_3|y_1, y_2)$ $\cdots$ $P(y_8|y_1, y_2, y_3, y_4, y_5, y_6, y_7)$



目標函數：

$$\prod_{t=1}^{n} P(y_t|y_1, y_2, \dots, y_{t-1})$$ ← Language Modeling

Self-attention 不可及的範圍

[Recap] Language Modeling and Cross-entropy

為了使語言模型能夠以分類的形式被訓練，**通常會取log**

$$\log\left(\prod_{t=1}^{n} P(y_t|y_1, y_2, \dots, y_{t-1})\right)$$

$$= \sum_{t=1}^{n} \log P(y_t|y_1, y_2, \dots, y_{t-1})$$ ← **加上負號之後就是 Cross-entropy**

(Recap) Perplexity

- Accuracy is not important for text generation.
- Perplexity 定義：language modeling 放在分母的幾何平均數
  - Why 放在分母？因為是困惑度，值越小越好，代表模型越有自信

在數學中，**幾何平均數**是一種均值，它通過使用它們的值的乘積（算術平均數使用"和"）來指示一組數字的集中趨勢或典型值。幾何平均數定義為第n根個數的乘積的第n個根，即對於一組數字$x_1, x_2, \ldots\ldots, x_n$，幾何平均數定義為：

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$

Perplexity:

$$\left(\prod_{t=1}^{n} \frac{1}{P(y_t|y_1, y_2, \ldots, y_{t-1})}\right)^{\frac{1}{n}}$$

Perplexity and geometric mean

以生成10個tokens為例 (前9個機率值都是0.9，最後一個機率值0.1):

算數平均數： $(0.9 * 9 + 0.0001)/10 \ = \ 0.81001$

幾何平均數： $\sqrt[10]{0.9^9 * 0.0001} = 0.3621$

BLEU (Bilingual Evaluation Understudy)

常用於機器翻譯

- A word-based metric.
  - It is very sensitive to word tokenization
- Core concept: Compute precision for n-grams:
  - Unigrams -> BLEU-1
  - Bigrams -> BLEU-2
  - Trigrams -> BLEU-3
  - 4-grams -> BLEU-4

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Precision and Recall

$$\text{Precision} = \frac{\text{Relevant and retrieved instances}}{\text{All retrieved instances}}$$

Predicted by a model

$$\text{Recall} = \frac{\text{Relevant and retrieved instances}}{\text{All relevant instances}}$$

Ground-truths

Relevant and retrieved instances: Intersection between predictions and ground-truths

Calculation of BLEU Score (Example)

機器翻譯常用兩個以上 references

Assume we now translate from Chinese to English.

Calculate BLEU-1 score

Chinese: 我想要讀那本書

Reference1: I want to read the book.

Reference2: I want to read that book.

Model output: the the the the the the.

Precision: $\frac{6}{6}$

100%! Can this be true?

Calculation of BLEU Score (Example)

Assume we now translate from Chinese to English.

Calculate BLEU-1 score

Chinese: 我想要讀那本書

Reference1: I want to read the book.

Reference2: I want to read that book.

Model output: the the the the the the.

~~Precision: $\frac{6}{6}$~~

Modified Precision: $\frac{1}{6}$

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Why should we use modified precision?

- The output sequences can be total mistakes.
  - E.g., the the the the the the
- Original precision is in favor of longer output sequences.
- Therefore, we should use modified precision to prevent bad evaluations.

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

1 2 3 4 5 6

	Count	
the dog	2	(duplicated)
dog the	1	
dog on	1	
on the	1	
the bed	1	

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

Clips to the reference

	Count	$\text{Count}_{\text{clip}}$
the dog	2	1
dog the	1	
dog on	1	
on the	1	
the bed	1	

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

	Count	Count_{clip}
the dog	2	1
dog the	1	0
dog on	1	
on the	1	
the bed	1	

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

	Count	Count_{clip}
the dog	2	1
dog the	1	0
dog on	1	1
on the	1	
the bed	1	

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

Count only one time even mapped to both references.

	Count	Count_{clip}
the dog	2	1
dog the	1	0
dog on	1	1
on the	1	1
the bed	1	

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Calculation of BLEU Score (Example)

Calculate BLEU-2 score

Reference1: The dog is on the bed.

Reference2: There is a dog on the bed.

Model output: The dog the dog on the bed.

Count only one time even mapped to both references.

	Count	$\text{Count}_{\text{clip}}$
the dog	2	1
dog the	1	0
dog on	1	1
on the	1	1
the bed	1	1

Modified Precision: $\frac{4}{6}$

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Formula of BLEU Score (1)

Summation for unigram, bigram, tri-gram, and 4-gram

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

Summation for all candidates (model outputs) of each translation

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Formula of BLEU Score (2)

Summation for unigram, bigram, tri-gram, and 4-gram

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

Summation for all candidates (model outputs) of each translation

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

What we've learned BLEU so far

- The BLEU score is calculated from the summation of 1-gram to 4-gram.
  - You can also measure n-gram individually.
- We use modified precision to prevent bad evaluations.
- What will happen if a model tends to generate really short sentences?

→ **More penalty for calculating BLEU score!**

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Brevity Penalty (BP)

- BP is used to penalize short candidates.

c: The length of a candidate sequence
r: The length of a reference sequence that is closest to c (shorter one)

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Then,

$$\text{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

N=4 to include 1-gram to 4-gram

Weight for each n-gram (was set 1/4 in the original paper)

Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

ROUGE Score

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation)
- Mainly for text summarization
- Metric Input: Summary (prediction), Reference (gold summary)
- Common metrics: ROUGE-1, ROUGE-2, ROUGE-L
    - L: Longest common subsequence
- Please note that current papers calculate ROUGE-F as default!!!
    - In other words, ROUGE-1F, ROUGE-2F, ROUGE-LF

Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text summarization branches out. 2004.

ROUGE-1 Example

predictions = ["The", "cat", "sat", "on", "the", "mat"]

references = ["A", "cat", "was", "sitting", "on", "the", "mat"]

ROUGE-1 recall = Number of matching unigrams / Number of unigrams in the reference = 4/7

ROUGE-1 precision = Number of matching unigrams / Number of unigrams in the machine-generated summary = 4/6

ROUGE-1 F1-score = Harmonic mean of the precision and the recall = 2 * 4/7 * 4/6 / (4/7 + 4/6)

ROUGE-2 Example

predictions = ["The cat", "cat sat", "sat on", "on the", "the mat"]

references = ["A cat", "cat was", "was sitting", "sitting on", "on the", "the mat"]

ROUGE-2 recall = Number of matching bigrams / Number of bigrams in the reference = 2/6

ROUGE-2 precision = Number of matching bigrams / Number of bigrams in the machine-generated summary = 2/5

ROUGE-2 F1-score = Harmonic mean of the precision and the recall = 2 * 2/6 * 2/5 / (2/6 + 2/5)

ROUGE-L Example

predictions = ["The", "cat", "sat", "on", "the", "mat"]

references = ["A", "cat", "was", "sitting", "on", "the", "mat"]

The longest common subsequence is ["cat", "on", "the", "mat"]

ROUGE-L recall = Number of matching unigrams / Number of unigrams in the reference = 4/7

ROUGE-L precision = Number of matching unigrams / Number of unigrams in the machine-generated summary = 4/6

ROUGE-L F1-score = Harmonic mean of the precision and the recall = 2 * 4/7 * 4/6 / (4/7 + 4/6)

ROUGE-L Example

The order should be kept for the LCS problem

predictions = ["The", "cat", "sat", "on", "the", "mat"]

references = ["on", "the", "mat”, "sitting", ”a", "cat"]

The longest common subsequence is ["on", "the", "mat"]

ROUGE-L recall = Number of matching unigrams / Number of unigrams in the reference = 3/6

ROUGE-L precision = Number of matching unigrams / Number of unigrams in the machine-generated summary = 3/6

ROUGE-L F1-score = Harmonic mean of the precision and the recall = 2 * 0.5 * 0.5 / (0.5 + 0.5)

Why do we need BLEU and ROUGE?

- BLEU is mainly designed for machine translation.
  - For example, the Brevity Penalty.
- ROUGE measures the overlapping between predicted and gold summaries.
- Can we just use one of them?
  - Conventionally, no.
  - Different tasks are evaluated with different metrics.

Comparison for Human and Automatic Evaluations (e.g., BLEU and ROUGE)

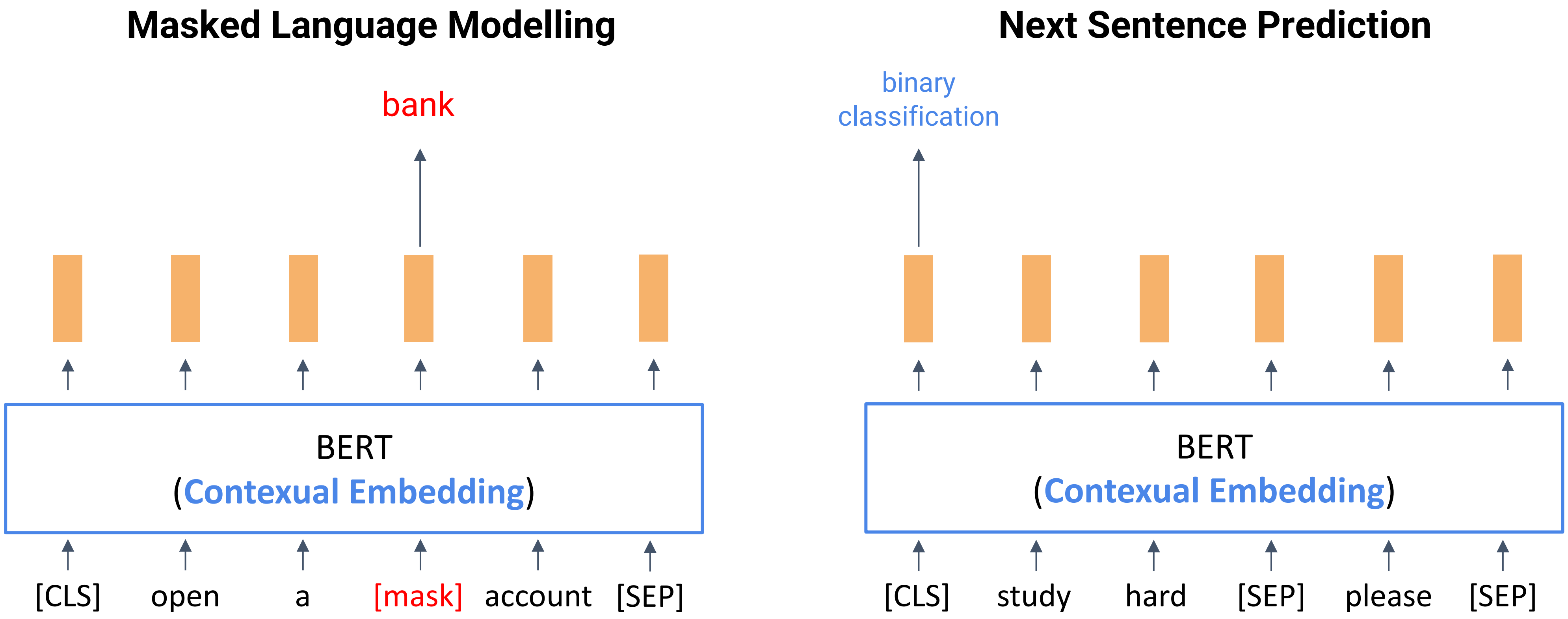
- Human evaluations
  - Pros: More accurate for subjectivity, flexibility for any desired comparison
  - Cons: Less objective, time-consuming, expensive
- Automatic evaluations
  - Pros: Objective enough to serve as common evaluation metrics, fast
  - Cons: Cannot meet language diversity
    - Take machine translation for instance, there are always other valid ways to translate the source sentence.

Issue of BLEU and ROUGE

- Cons: Cannot meet language diversity
  - This mainly comes from the way for measuring overlapping rates.
- **Question**: Can we create an automatic metric to fix the issue?

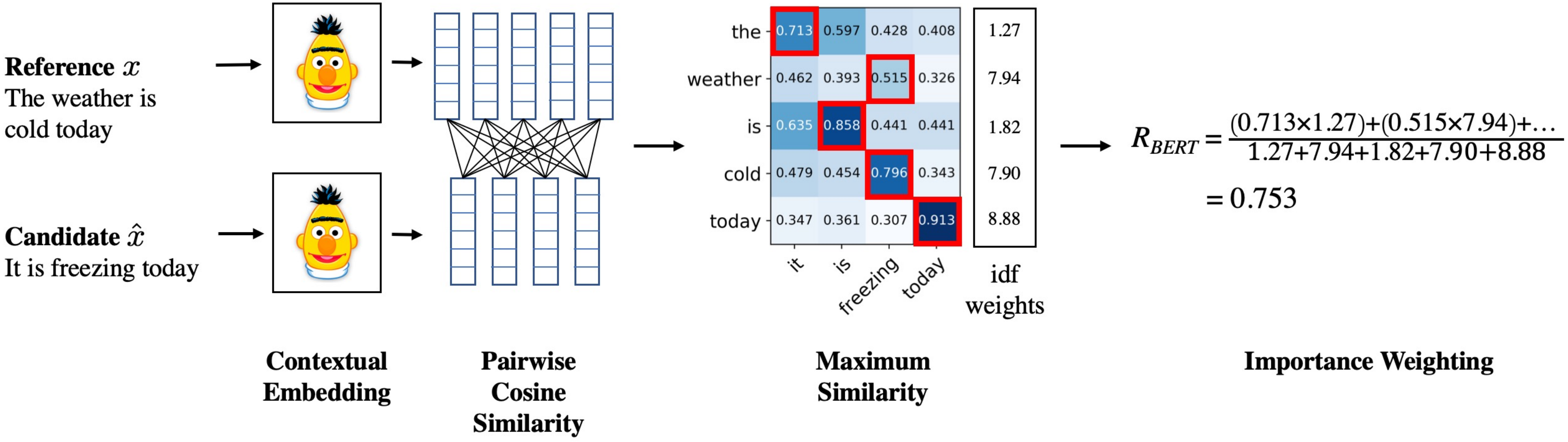
- Next, we are going to introduce two learned automatic evaluation metrics
  - BERTScore (ICLR 2020)
  - BLEURT (ACL 2020)

(Recap) BERT: Bidirectional Encoder Representations from Transformers



Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL 2019.

BERTScore – Overview



Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." International Conference on Learning Representations. 2020.

BERTScore – Steps

Step 0: Prepare Reference x, Candidate $\hat{x}$, and a pre-trained BERT model

Step 1: Infer x and $\hat{x}$ with BERT respectively, get a sequence of output vectors $\langle \mathrm{x}_1, \ldots, \mathrm{x}_k \rangle$ for x and a sequence of output vectors $\langle \hat{\mathrm{x}}_1, \ldots, \hat{\mathrm{x}}_k \rangle$ for $\hat{x}$

Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." International Conference on Learning Representations. 2020.

BERTScore – Steps

Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." International Conference on Learning Representations. 2020.

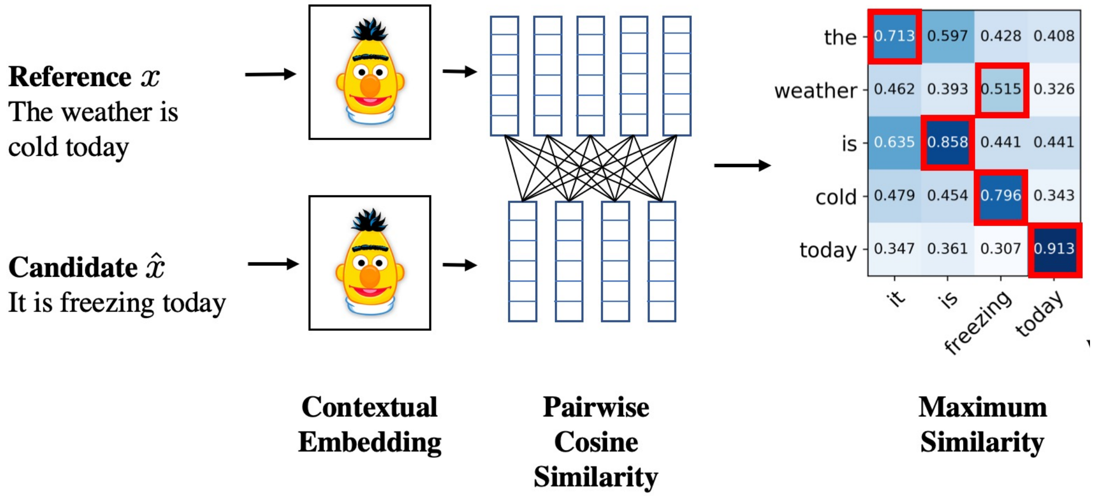
Step 0: Prepare Reference x, Candidate $\hat{x}$, and a pre-trained BERT model

Step 1: Infer x and $\hat{x}$ with BERT respectively, get a sequence of output vectors $\langle \mathrm{x}_1, \ldots, \mathrm{x}_k \rangle$ for x and a sequence of output vectors $\langle \hat{\mathrm{x}}_1, \ldots, \hat{\mathrm{x}}_k \rangle$ for $\hat{x}$

Step 2: Measure pairwise cosine similarity

Recall
$$R_{\mathrm{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\boxed{\hat{x}_j \in \hat{x}}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$
Based on reference

Precision
$$P_{\mathrm{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{\boxed{x_i \in x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j$$
Based on candidate

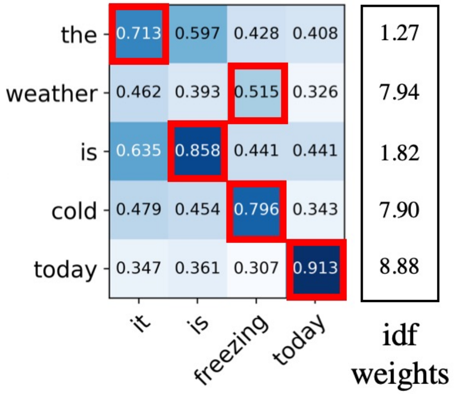


BERTScore – Importance Weighting

Given M reference sentences $\{x^{(i)}\}_{i=1}^{M}$, the idf (inverse document frequency) score of a word-piece token w is:

$$\mathrm{Idf}(w) = -\log \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[w \in x^{(i)}\right]$$

$$R_{\mathrm{BERT}} = \frac{\sum_{x_i \in x} \mathrm{idf}(x_i) \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^{\top} \hat{\mathbf{x}}_j}{\sum_{x_i \in x} \mathrm{idf}(x_i)}$$



$$\longrightarrow \quad R_{BERT} = \frac{(0.713\times1.27)+(0.515\times7.94)+\ldots}{1.27+7.94+1.82+7.90+8.88}$$
$$= 0.753$$

Maximum Similarity

Importance Weighting

Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." International Conference on Learning Representations. 2020.

Summary of BERTScore

- BERTScore leverages the contextual representation abilities of BERT to measure the semantic similarities between a reference and a candidate.
- In the paper, BERTScore correlates better with human judgments and provides stronger model selection performance than existing metrics.
- However, BERTScore does not involve training process.

Can we train BERT for a better evaluation metric?

Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with BERT." International Conference on Learning Representations. 2020.

BLEURT – Quick Introduction

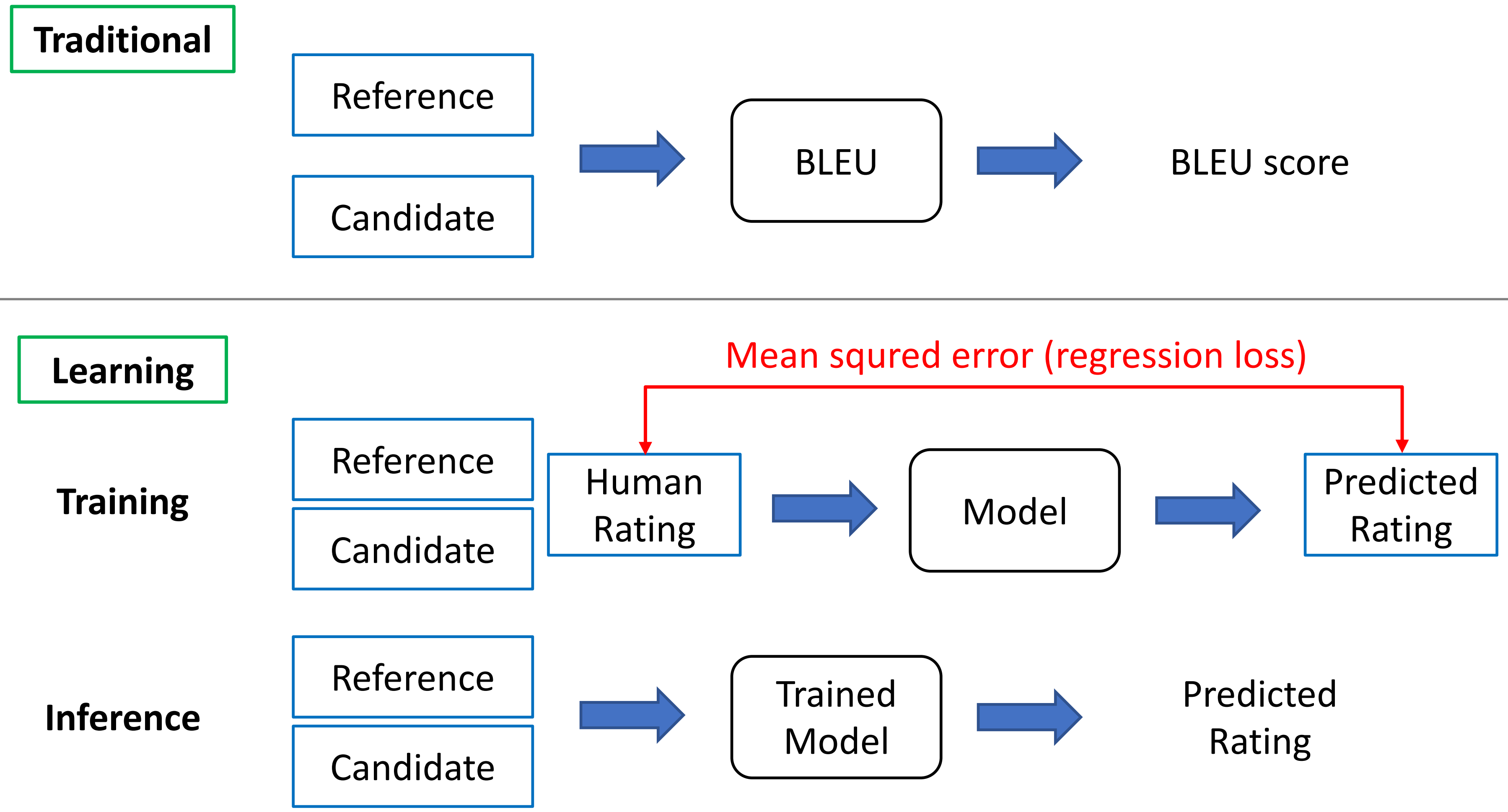
- BLEURT: Learning Robust Metrics for Text Generation, published by Google
- BLEURT trains BERT for a more robust evaluation metric.
  - Mainly for machine translation.
    - Also get hints from the name "BLEURT"
  - Trained checkpoint can be obtained. We don't need to perform training.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

BLEURT – Motivations

- Learned metrics can be tuned to measure task-specific properties, such as fluency, faithfulness, grammar, or style.
- NLG systems tend to get better over time, and therefore a model trained on ratings data from 2015 may fail to distinguish top performing systems in 2019, especially for newer research tasks.

Training on Human Ratings



BLEURT – Steps

Step 0: Reference-candidate pairs $(z, \tilde{z})$ and the pre-trained BERT model

Step 1: Data augmentation for $(z, \tilde{z})$ to to perform pre-training

Data augmentation strategies

- Random masking
- Back-translation
- Dropping words randomly

Total 6.5 million variants of $(z, \tilde{z})$ were created.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

Random masking

Two kinds of masking strategies were adopted:

Token masking

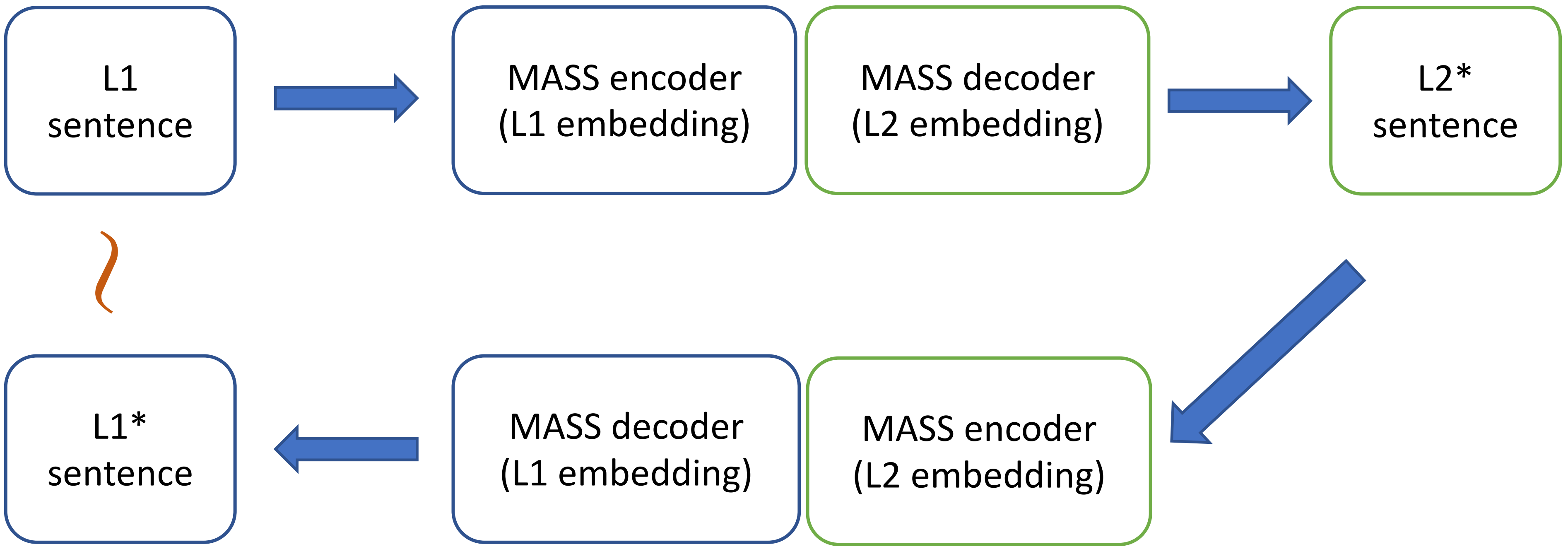
I love traveling to Vancouver for attending a conference.	→	I love traveling to Vancouver for [MASK] a conference.

Span masking

I love traveling to Vancouver for attending a conference.	→	I love traveling to Vancouver for [MASK] [MASK] [MASK].

Backtranslation

- L1: English; L2: French or German



Dropping words randomly

- The authors found it useful in their experiments to randomly drop words to create other examples.

I love traveling to Vancouver for attending a conference.	→	I love to Vancouver for attending a conference.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

BLEURT – Step 3

Step 3: Pre-training each sentence pair $(z, \tilde{z})$ with the following tasks.

Note that this is not conventional BERT pre-training! It is multi-task pre-training!

Task Type	Pre-training Signals	Loss Type
BLEU	$\boldsymbol{\tau}_{\text{BLEU}}$	Regression
ROUGE	$\boldsymbol{\tau}_{\text{ROUGE}} = (\tau_{\text{ROUGE-P}}, \tau_{\text{ROUGE-R}}, \tau_{\text{ROUGE-F}})$	Regression
BERTscore	$\boldsymbol{\tau}_{\text{BERTscore}} = (\tau_{\text{BERTscore-P}}, \tau_{\text{BERTscore-R}}, \tau_{\text{BERTscore-F}})$	Regression
Backtrans. likelihood	$\boldsymbol{\tau}_{\text{en-fr},\boldsymbol{z}\mid\tilde{\boldsymbol{z}}}, \boldsymbol{\tau}_{\text{en-fr},\tilde{\boldsymbol{z}}\mid\boldsymbol{z}}, \boldsymbol{\tau}_{\text{en-de},\boldsymbol{z}\mid\tilde{\boldsymbol{z}}}, \boldsymbol{\tau}_{\text{en-de},\tilde{\boldsymbol{z}}\mid\boldsymbol{z}}$	Regression
Entailment	$\boldsymbol{\tau}_{\text{entail}} = (\tau_{\text{Entail}}, \tau_{\text{Contradict}}, \tau_{\text{Neutral}})$	Multiclass
Backtrans. flag	$\boldsymbol{\tau}_{\text{backtran\_flag}}$	Multiclass

- Ground-truth values can be computed for each $(z, \tilde{z})$ pair!
- Losses for the six tasks were sum up during pre-training.

- Regression: mean squared error
- Multiclass: Cross-entropy

Sellam, Thibault, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

Task 4: Backtranslation Likelihood

- Existing translation models (trained) are needed.
  - Transformers (Vaswani et al., 2017): EN-FR
  - Transformers (Vaswani et al., 2017): DE-EN
- Equations use EN-FR for an example

$P(x_t | x_1, \ldots, x_{t-1}, z)$

$$\boldsymbol{z}^*_{\mathtt{fr}} = \arg\max P_{\mathtt{en}\rightarrow\mathtt{fr}}(\boldsymbol{z}_{\mathtt{fr}} | \boldsymbol{z})$$

Best translated French sentence (details absent in the paper)

$$P(\tilde{\boldsymbol{z}} | \boldsymbol{z}) \approx P_{\mathtt{fr}\rightarrow\mathtt{en}}(\tilde{\boldsymbol{z}} | \boldsymbol{z}^*_{\mathtt{fr}})$$

Backtranslation Likelihood

Task 5 and Task 6

- **Textual Entailment**
  - We report the probability of three labels: Entail, Contradict, and Neutral, using BERT fine-tuned on the MNLI dataset.
- **Backtranslation flag**
  - A Boolean that indicates whether the perturbation was generated with backtranslation or with mask-filling.

BLEURT – Final Step

Step 4: Fine-tune the model (trained from Step 3) on the <Reference, Candidate ,Rating> data using the regression loss

The <Reference, Candidate ,Rating> data include

- WMT (machine translation task)
- WebNLG (for general text generation)

Summary of BLEURT

- This approach uses (continual) pre-training and fine-tuning to create a learned evaluation metric for machine translation and general NLG.
- According to the paper, BLEURT is better aligned to human ratings then BERTScore.
- BLEURT should work for text summarization, but the authors did not test it.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh. "BLEURT: Learning Robust Metrics for Text Generation." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

Thank you!

Instructor: 林英嘉

✉️ yjlin@cgu.edu.tw