Common Evaluation Metrics for NLP

Faith, Halimat, Shaheen, Bachar

Perplexity (PPL): how surprised the model is

Uncertainty in the underlying probability distribution

Evaluate performance and compare probabilistic models: intrinsic, fast

GOAL: minimise perplexity, maximise probability. The LOWER perplexity corresponds the more confident and better performing the model

e.g. I always order pizza with…mushrooms (0.1), pepperoni (0.1), anchovies(0.01), fried rice (0.0001)

How well it predicts the actual word that occurs. Assigns a higher probability to the word that actually occurs

Probability related to the entropy of upcoming things…

PPL as Branching Factor

How many things can occur at each time weighted by their probability: normalising the probability of a long string.

Take the weighted average of all possibilities to compute on average how likely any one word can occur.

e.g. System: Operator (1 in 4), Sales (1 in 4), Tech Support (1 in 4), 30,000 names (1 in 120,000 each), Perplexity is 54

Mulitply 120k probabilite (90k are ¼, and 30k are 1/120k) and take the inverse of 120,00th root

Perp = (¼*¼*¼*1/120)^(-¼) = 52.6

Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

BLEU

- Evaluating Machine Translation
- Human translations available
- Compare machine vs human translation

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

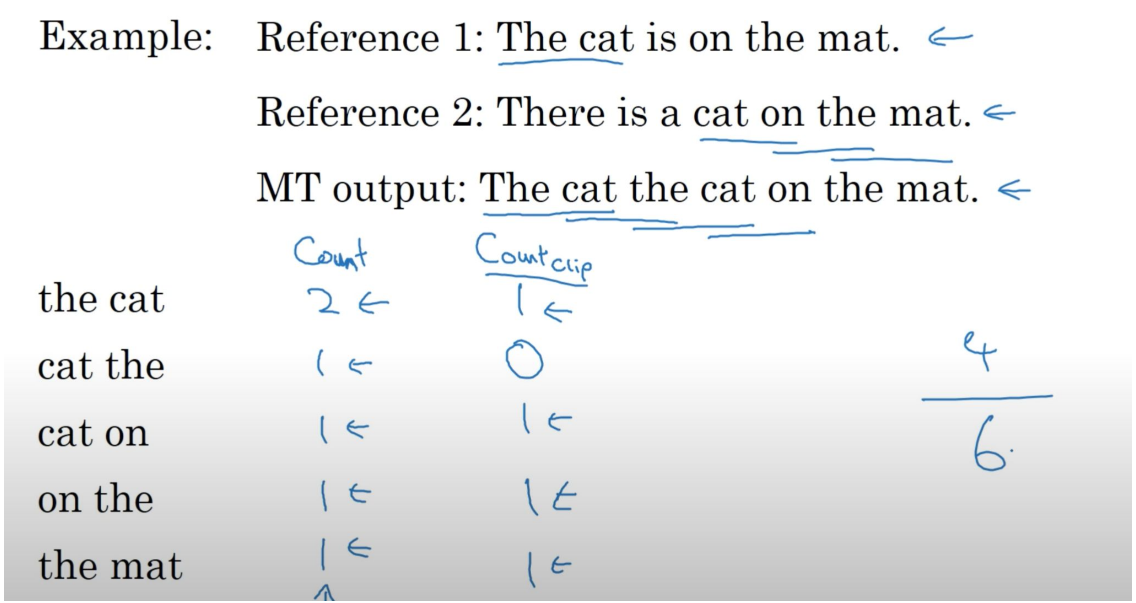
MT output: the the the the the the the.

1. Precision - how many MT words in HT? 7/7
2. Modified Precision - Credit up to max occurrences - 1/7 or 2/7

But what about pairs of words?

BLEU - Bigrams

1. List all bigrams in MT
2. Count occurrences of each bigram
3. Clip occurrences by max(occurrences) in one training sentence
4. Modified bigram precision:
    a. Sum(Counts Clipped) / Sum(Counts)



BLEU - N-Grams

1. Calculate Modified Precision for unigram, bigram, trigram, quadgram etc.
2. BLEU = (Brevity Penalty) x EXP(Sum of modified precision scores)
3. Brevity Penalty:
   a. = 1 if MT longer than HT
   b. = < 1 if MT shorter than HT

GLEU

- BLEU doesn't work well for single sentences
- Recall (for unigram/bigram/trigram etc):
  - N_matching(MT to HT) / N_total in HT
- Precision (for unigram/bigram/trigram etc):
  - N_matching(MT to HT) / N_total in MT

- GLEU_sentence = MIN(Recall, Precision)
- GLEU_corpus = AVG(GLEU_sentences)

METEOR - What does it do?

METEOR (Metric for Evaluation of Translation with Explicit ORdering) is a machine translation evaluation metric.

Multiple `references` per `prediction`:

```
>>> meteor = evaluate.load('meteor')
>>> predictions = ["It is a guide to action which ensures that the military always obeys the commands of the party"]
>>> references = [['It is a guide to action that ensures that the military will forever heed Party commands', 'It is the guiding principle which guarantees the militar
>>> results = meteor.compute(predictions=predictions, references=references)
>>> print(round(results['meteor'], 2))
1.0
```

Multiple `references` per `prediction`, partial match:

```
>>> meteor = evaluate.load('meteor')
>>> predictions = ["It is a guide to action which ensures that the military always obeys the commands of the party"]
>>> references = [['It is a guide to action that ensures that the military will forever heed Party commands', 'It is the guiding principle which guarantees the militar
>>> results = meteor.compute(predictions=predictions, references=references)
>>> print(round(results['meteor'], 2))
0.69
```

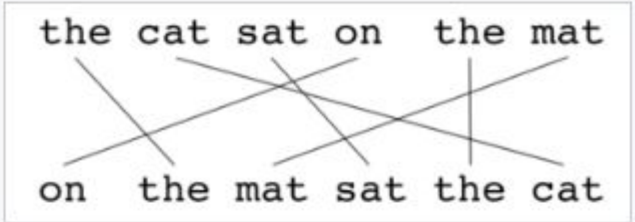
METEOR - Motivation Behind Developing it

- **Has several features that are not found in other metrics**, such as **stemming and synonymy matching**, along with the standard exact word matching.
- **Fix** some of the **problems** found in the more popular **BLEU** metric
- **Produce good correlation with human judgement at the sentence or segment level.** BLEU seeks correlation at the corpus level.

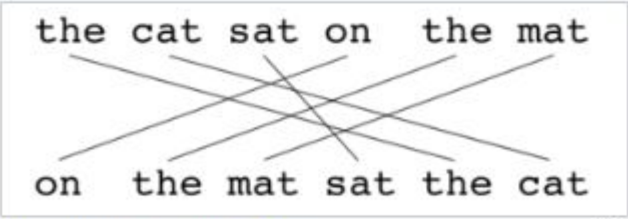
Examples of pairs of words which will be mapped by each module

Module	Candidate	Reference	Match
Exact	Good	Good	Yes
Stemmer	Goods	Good	Yes
Synonymy	well	Good	Yes

METEOR - Under the Hood



Example alignment (a).



Example alignment (b).

$$P = \frac{m}{w_t}$$

$$R = \frac{m}{w_r}$$

$$F_{mean} = \frac{10PR}{R + 9P}$$

recall weighted 9 times more than precision to value **completeness over correctness.**

$$p = 0.5\left(\frac{c}{u_m}\right)^3$$

The more mappings there are that are **not adjacent in the reference and the candidate sentence**, the higher the penalty will be

METEOR - Example

Reference	the	cat	sat	on	the	mat
Hypothesis	the	cat	sat	on	the	mat
Score	$0.9977 = \underbrace{1.0000}_{\text{Fmean}} \times (1 - \underbrace{0.0023}_{\text{Penalty}})$					
Fmean	$1.0000 = 10 \times \underbrace{1.0000}_{\text{Precision}} \times \frac{\overbrace{1.0000}^{\text{Recall}}}{\underbrace{1.0000}_{\text{Recall}} + 9 \times \underbrace{1.0000}_{\text{Precision}}}$					
Penalty	$0.0023 = 0.5 \times \underbrace{0.1667^3}_{\text{Fragmentation}}$					
Fragmentation	$0.1667 = \frac{\overbrace{1.0000}^{\text{Chunks}}}{\underbrace{6.0000}_{\text{Matches}}}$					

Reference	the	cat	sat	on	the	mat
Hypothesis	on	the	mat	sat	the	cat
Score	$0.9375 = \underbrace{1.0000}_{\text{Fmean}} \times (1 - \underbrace{0.0625}_{\text{Penalty}})$					
Fmean	$1.0000 = 10 \times \underbrace{1.0000}_{\text{Precision}} \times \frac{\overbrace{1.0000}^{\text{Recall}}}{\underbrace{1.0000}_{\text{Recall}} + 9 \times \underbrace{1.0000}_{\text{Precision}}}$					
Penalty	$0.0625 = 0.5 \times \underbrace{0.5^3}_{\text{Fragmentation}}$					
Fragmentation	$0.5 = \frac{\overbrace{3.0000}^{\text{Chunks}}}{\underbrace{6.0000}_{\text{Matches}}}$					

METEOR - So when to use it

- Translation task
- Care about completion not just precision
- Care about precision and completion at the sentence and segment level

Disclaimer: This is my understanding

ROUGE - Recall-Oriented Understudy for Gisting Evaluation What does it do?

Used for evaluating **automatic summarization and machine translation** software in natural language processing.

The metrics compare an **automatically produced** summary or translation against a reference or a set of references **(human-produced)** summary or translation.

Recall: This is like the robot's memory.
Gisting: Gisting is a fancy word for getting the main idea.
Evaluation: This is like giving a report card.

ROUGE - Recall-Oriented Understudy for Gisting Evaluation
How it does it?

ROUGE-1 ('rouge1') - **Recall**: does the generated summary contain all the important **words (unigrams)**

ROUGE-2 ('rouge2') - **Recall**: does the generated summary contain all the important **two-word phrase (bigrams)**

ROUGE-L ('rougeL') - **Recall + Order**: does the generated summary contain the **main points and in the right order** (**flow of the story)**

ROUGE-Lsum ('rougeLsum') **Recall + Order + Precision**: does the generated summary have **similar structure and meaningful words**

Model output: "A fast brown fox leaps over a sleeping dog."

Rouge-L

Reference summary: "The quick brown fox jumps over the lazy dog."

ROUGE - Recall-Oriented Understudy for Gisting Evaluation

Example?

Reference Text: "The quick brown fox jumps over the lazy dog."

Generated Summary Text: "A fast brown fox leaps over a sleeping dog."

ROUGE-1

- Common Unigrams: ['brown', 'fox', 'over']
- Precision (Common / Summary): 3 / 9 = 0.3333
- Recall (Common / Reference): 3 / 9 = 0.3333
- F1 Score (Harmonic Mean): 2 * (0.3333 * 0.3333) / (0.3333 + 0.3333) = 0.3333

ROUGE-2

- Reference Bigrams: ['The quick', 'quick brown', 'brown fox', 'fox jumps', 'jumps over', 'over the', 'the lazy', 'lazy dog']
- Summary Bigrams: ['A fast', 'fast brown', 'brown fox', 'fox leaps', 'leaps over', 'over a', 'a sleeping', 'sleeping dog']
- Common Bigrams: ['brown fox']
- Precision (Common / Summary): 1 / 8 = 0.125
- Recall (Common / Reference): 1 / 8 = 0.125
- F1 Score (Harmonic Mean): 2 * (0.125 * 0.125) / (0.125 + 0.125) = 0.125

ROUGE-L

- Longest Common Subsequence (LCS): ['brown', 'fox', 'over']
- Reference Length: 9 words
- Precision (LCS Length / Summary Length): 3 / 9 = 0.3333
- Recall (LCS Length / Reference Length): 3 / 9 = 0.3333
- F1 Score (Harmonic Mean): 2 * (0.3333 * 0.3333) / (0.3333 + 0.3333) = 0.3333

ROUGE - So when to use it

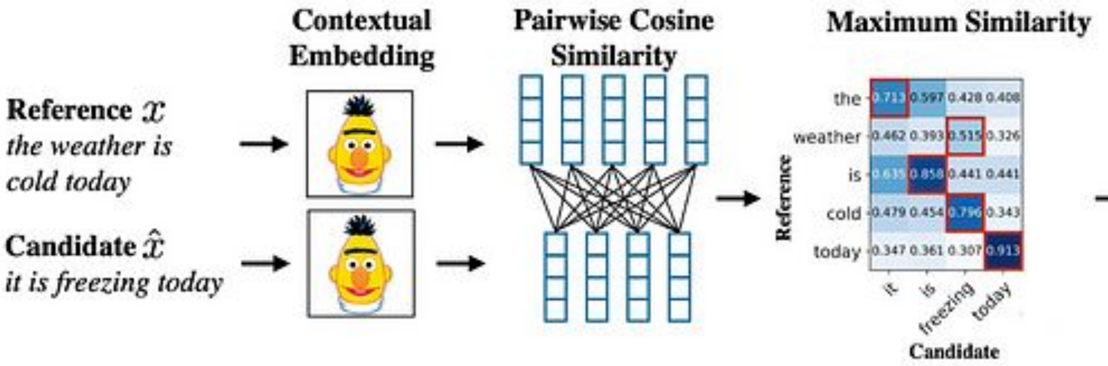
- Summarization/Translation task
- Care about completeness at the word/two-word level (maybe keywords/scientific documents)
- Care about capturing the main idea and the flow of the narrative/structure

Disclaimer: This is my understanding

BERTScore

- BERTScore is an automatic evaluation metric used for testing the goodness of text generation systems.
- Unlike existing popular methods that compute token level syntactical similarity, BERTScore focuses on computing semantic similarity between tokens of reference and hypothesis.
- The author's of the paper tested it on machine translations and image captioning tasks and found it to correlate better with human judgements.

Introducing **BERTScore**



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

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

$$F_{\mathrm{BERT}} = 2 \frac{P_{\mathrm{BERT}} \cdot R_{\mathrm{BERT}}}{P_{\mathrm{BERT}} + R_{\mathrm{BERT}}}$$

Source: Bertscore: Evaluating text generation with bert

Code for Bertscore is available at https://github.com/Tiiiger/bert score