Natural language processing
Different tasks

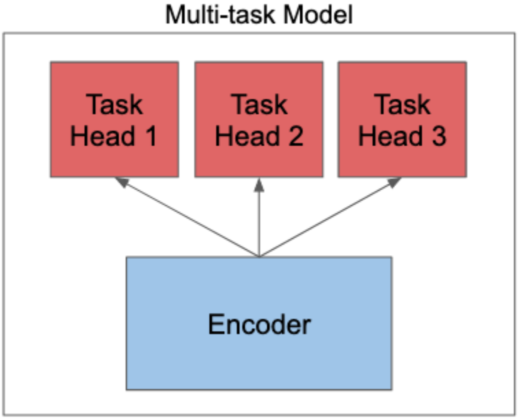
Linas Petkevičius, PhD

Institute of Computer Science
Vilnius University
linas.petkevicius@mif.vu.lt

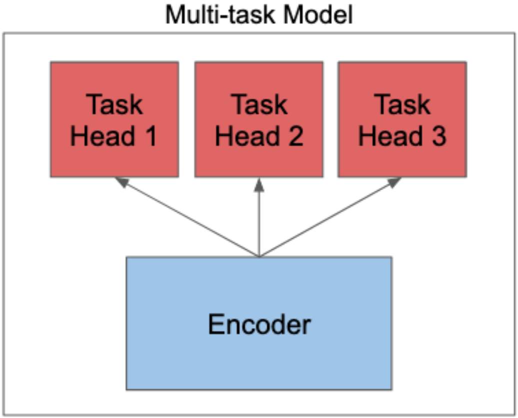
30^{th} November 2022

Today

- Team project
- Measures for readability


Multi-task Model
Task Head 1
Task Head 2
Task Head 3
Encoder

Multi-task learning



Example

Alternative: Finetuning model using custom loss

```
from transformers import Trainer

class BartTrainer(Trainer):
    def compute_loss(self, model, inputs):
        # implement custom logic here
        custom_loss = ...
        return custom_loss
```

Perplexity

Perplexity is a measurement of how well a probability distribution or probability model predicts a sample.

$$\text{Perplexity}(M) = M(s)^{-1/n}$$

$$= \sqrt[n]{\prod_{k=1}^{n} \frac{1}{M(w_k | w_0 w_1 \cdots w_{k-1})}}$$

Bilingual Evaluation Understudy Score (BLEU)

Mathematically, the BLEU score is defined as:

$$\text{BLEU} = \underbrace{\min\Big(1, \exp\big(1 - \frac{\text{reference-length}}{\text{output-length}}\big)\Big)}_{\text{brevity penalty}} \underbrace{\Big(\prod_{i=1}^{4} precision_i\Big)^{1/4}}_{\text{n-gram overlap}}$$

with

$$precision_i = \frac{\sum_{\text{snt}\in\text{Cand-Corpus}} \sum_{i\in\text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'}\in\text{Cand-Corpus}} \sum_{i'\in\text{snt'}} m^{i'}_{cand}}$$

where

- m^i_{cand} is the count of i-gram in candidate matching the reference translation
- m^i_{ref} is the count of i-gram in the reference translation
- w^i_t is the total number of i-grams in candidate translation

BLEU

In [9]:
```
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']
score = sentence_bleu(reference, candidate)
print(score)
```

```
1.0
```

In [10]:
```
# one word different
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'fast', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']
score = sentence_bleu(reference, candidate)
print(score)
```

```
0.7506238537503395
```

In [11]:
```
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'fast', 'brown', 'fox', 'jumped', 'over', 'the', 'sleepy', 'dog']
score = sentence_bleu(reference, candidate)
print(score)
```

```
0.4854917717073234
```

BLEU

In [12]:
```python
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
score = sentence_bleu(reference, candidate)
print(score)
```

```
0
```

In [13]:
```python
# shorter candidate
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the']
score = sentence_bleu(reference, candidate)
print(score)
```

```
0.7514772930752859
```

Examples

Calculating $precision_1$

Consider this reference sentence and candidate translation:

Reference: `the cat is on the mat`
Candidate: `the the the cat mat`

The first step is to count the occurrences of each unigram in the reference and the candidate. Note that the BLEU metric is case-sensitive.

Unigram	m^i_{cand}	m^i_{ref}	$\min(m^i_{cand}, m^i_{ref})$
`the`	3	2	2
`cat`	1	1	1
`is`	0	1	0
`on`	0	1	0
`mat`	1	1	1

The total number of unigrams in the candidate (w^1_t) is 5, so $precision_1$ = (2 + 1 + 1)/5 = 0.8.

Calculating the BLEU score

Reference: `The NASA Opportunity rover is battling a massive dust storm on Mars .`
Candidate 1: `The Opportunity rover is combating a big sandstorm on Mars .`
Candidate 2: `A NASA rover is fighting a massive storm on Mars .`

The above example consists of a single reference and two candidate translations. The sentences are tokenized prior to computing the BLEU score as depicted above; for example, the final period is counted as a separate token.

To compute the BLEU score for each translation, we compute the following statistics.

- **N-Gram Precisions**
  The following table contains the n-gram precisions for both candidates.
- **Brevity-Penalty**
  The brevity-penalty is the same for candidate 1 and candidate 2 since both sentences consist of 11 tokens.
- **BLEU-Score**
  Note that at least one matching 4-gram is required to get a BLEU score > 0. Since candidate translation 1 has no matching 4-gram, it has a BLEU score of 0.

Metric	Candidate 1	Candidate 2
$precision_1$ (1gram)	8/11	9/11
$precision_2$ (2gram)	4/10	5/10
$precision_3$ (3gram)	2/9	2/9
$precision_4$ (4gram)	0/8	1/8
Brevity-Penalty	0.83	0.83
BLEU-Score	**0.0**	**0.27**

- N-gram Co-Occurrence Statistics (ROUGE-N)

[1]

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

ROUGE-N

$$= \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

[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.

Q & A