Natural Language Processing

CS 3216/UG, AI 5203/PG

Week-10

BLEU Score, Introduction to LLM's

Mahindra™ University

Global Thinkers. Engaged Leaders.

Recap

- Language modeling
- Recurrent Neural Network and Implementation
- Applications of Recurrent Neural Network
- Language modeling using Long Short-term Memory
- Sequence to Sequence learning
- Attention
- Transformers

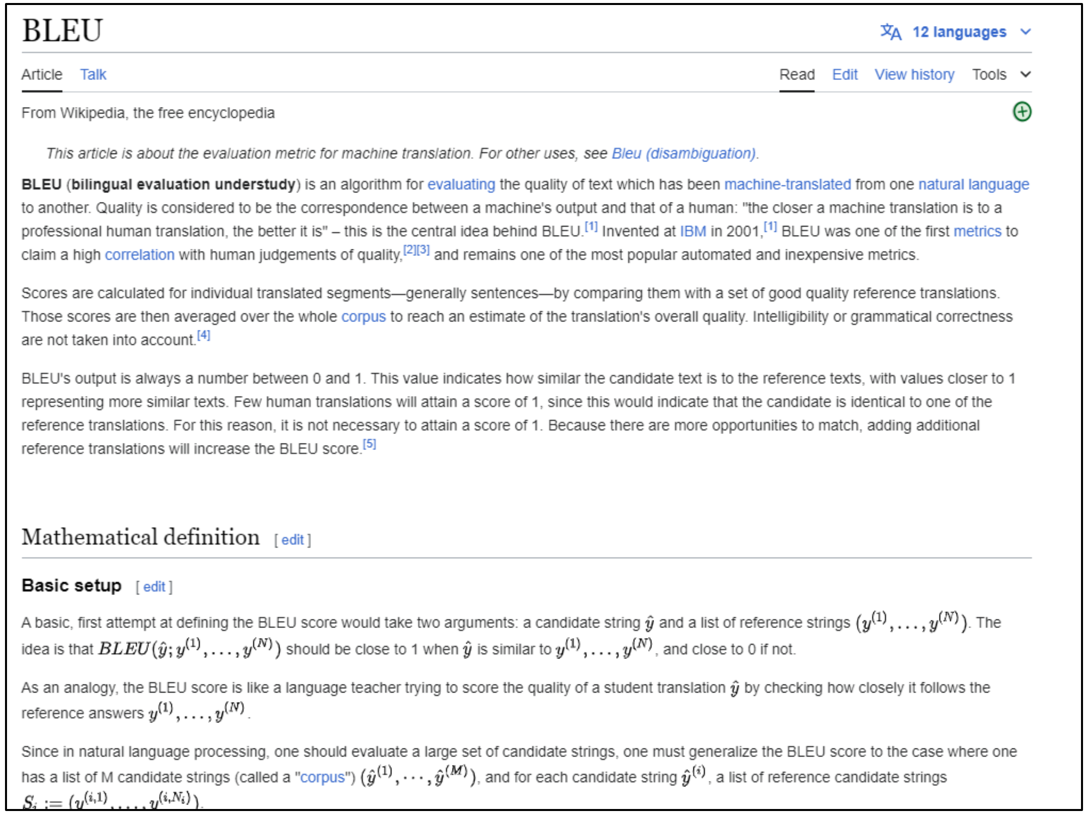
Why?

- You covered Transformers for Machine Translation Task.

How would you evaluate the quality of translations?

How to evaluate Machine Translation?

BLEU (Bilingual Evaluation Understudy)



BLEU

From Wikipedia, the free encyclopedia

This article is about the evaluation metric for machine translation. For other uses, see Bleu (disambiguation).

BLEU (**bilingual evaluation understudy**) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU.[1] Invented at IBM in 2001,[1] BLEU was one of the first metrics to claim a high correlation with human judgements of quality,[2][3] and remains one of the most popular automated and inexpensive metrics.

Scores are calculated for individual translated segments—generally sentences—by comparing them with a set of good quality reference translations. Those scores are then averaged over the whole corpus to reach an estimate of the translation's overall quality. Intelligibility or grammatical correctness are not taken into account.[4]

BLEU's output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts. Few human translations will attain a score of 1, since this would indicate that the candidate is identical to one of the reference translations. For this reason, it is not necessary to attain a score of 1. Because there are more opportunities to match, adding additional reference translations will increase the BLEU score.[5]

Mathematical definition [edit]

Basic setup [edit]

A basic, first attempt at defining the BLEU score would take two arguments: a candidate string $\hat{y}$ and a list of reference strings $(y^{(1)}, \dots, y^{(N)})$. The idea is that $BLEU(\hat{y}; y^{(1)}, \dots, y^{(N)})$ should be close to 1 when $\hat{y}$ is similar to $y^{(1)}, \dots, y^{(N)}$, and close to 0 if not.

As an analogy, the BLEU score is like a language teacher trying to score the quality of a student translation $\hat{y}$ by checking how closely it follows the reference answers $y^{(1)}, \dots, y^{(N)}$.

Since in natural language processing, one should evaluate a large set of candidate strings, one must generalize the BLEU score to the case where one has a list of M candidate strings (called a "corpus") $(\hat{y}^{(1)}, \cdots, \hat{y}^{(M)})$, and for each candidate string $\hat{y}^{(i)}$, a list of reference candidate strings $S_i := (y^{(i,1)}, \dots, y^{(i,N_i)})$.

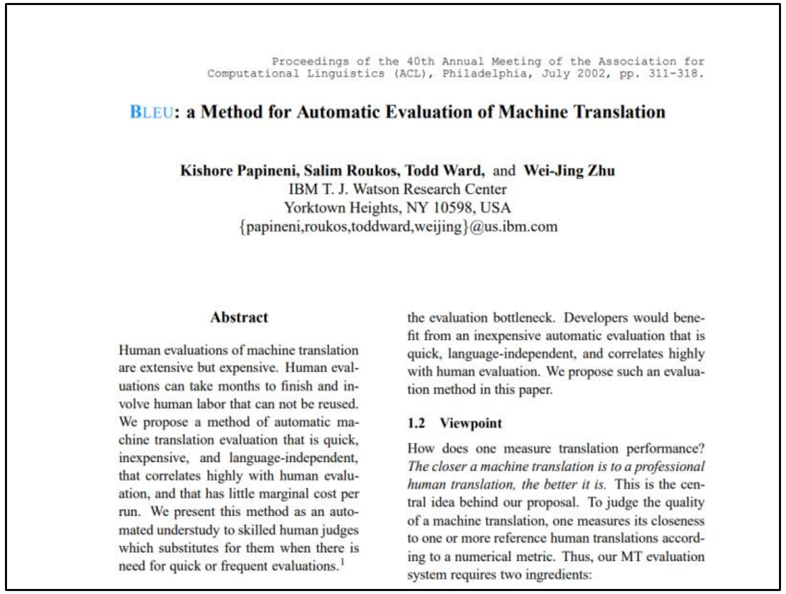
https://en.wikipedia.org/wiki/BLEU

How to evaluate Machine Translation?

BLEU (Bilingual Evaluation Understudy)

BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:

- Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
- + a penalty for too-short system translations



Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

BLEU: a Method for Automatic Evaluation of Machine Translation

Kishore Papineni, Salim Roukos, Todd Ward, and **Wei-Jing Zhu**
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
{papineni,roukos,toddward,weijing}@us.ibm.com

Abstract

Human evaluations of machine translation are extensive but expensive. Human evaluations can take months to finish and involve human labor that can not be reused. We propose a method of automatic machine translation evaluation that is quick, inexpensive, and language-independent, that correlates highly with human evaluation, and that has little marginal cost per run. We present this method as an automated understudy to skilled human judges which substitutes for them when there is need for quick or frequent evaluations.[1]

the evaluation bottleneck. Developers would benefit from an inexpensive automatic evaluation that is quick, language-independent, and correlates highly with human evaluation. We propose such an evaluation method in this paper.

1.2 Viewpoint

How does one measure translation performance? *The closer a machine translation is to a professional human translation, the better it is.* This is the central idea behind our proposal. To judge the quality of a machine translation, one measures its closeness to one or more reference human translations according to a numerical metric. Thus, our MT evaluation system requires two ingredients:

Source paper:*"BLEU: a Method for Automatic Evaluation of Machine Translation", Papineni et al, 2002*

https://aclanthology.org/P02-1040.pdf

How to calculate BLEU Score?

French: Le chat est sur le tapis → Source language

Reference: The cat is on the mat → Human-Written Translation

Model

MT Output- Candidate: The cat on mat — Machine generated Translation

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

m = No of candidate words in the reference

w_t = Total No of words in the candidate

$m = 4$

$W_t = 4$

$$= \frac{4}{4} \boxed{= 1}$$

How to calculate BLEU Score? (What can go wrong?)

French: Le chat est sur le tapis

Reference 1: The cat is on the mat

Reference 2: There is a cat on the mat

Reference

MT Output- Candidate: the the the the the the the

$m =$ No of candidate words in the refer

$m = 7$

$w_t = 7$

$$P = \frac{m}{w_t} = \frac{7}{7} = 1$$

How to calculate BLEU Score?

MT The cat is on the mat

MT the cat the cat on the mat

	Count	Count clip
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1

Modified

$p = \frac{7}{7} = 1$

m_t w_t

$= \frac{3}{6}$

$m_t = 7$ $w_t = 7$

200 years

→ Bigram

My Neural Machine Translation model:
generates some output

BLEU Score:

How to calculate BLEU Score?

BLEU Score

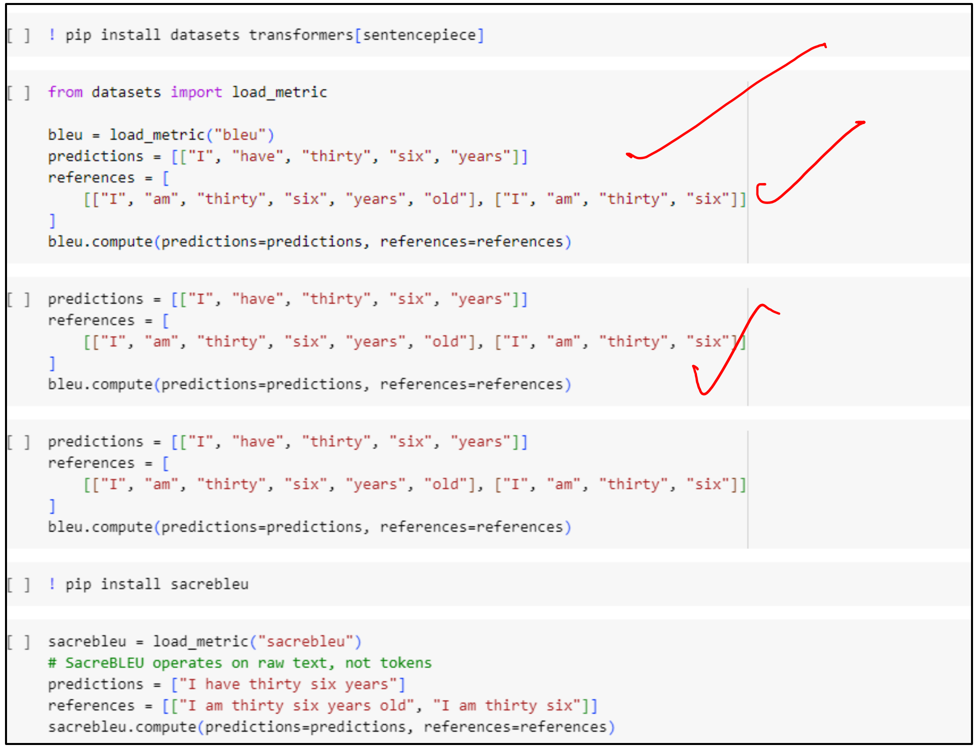
Pros:

- Simple and widely used metric
- BLEU is useful but imperfect

Cons:

- Doesn't incorporate sentence structure.
  - If the word is placed in the wrong place then meaning changes
- Does not take in account semantic similarity of the words
  - There are many valid ways to translate a sentence
- More focused on precision rather than recall
- Struggle with non-english languages.

BLEU Score Python Code



```
[ ]  ! pip install datasets transformers[sentencepiece]
```

```
[ ]  from datasets import load_metric

     bleu = load_metric("bleu")
     predictions = [["I", "have", "thirty", "six", "years"]]
     references = [
         [["I", "am", "thirty", "six", "years", "old"], ["I", "am", "thirty", "six"]]
     ]
     bleu.compute(predictions=predictions, references=references)
```

```
[ ]  predictions = [["I", "have", "thirty", "six", "years"]]
     references = [
         [["I", "am", "thirty", "six", "years", "old"], ["I", "am", "thirty", "six"]]
     ]
     bleu.compute(predictions=predictions, references=references)
```

```
[ ]  predictions = [["I", "have", "thirty", "six", "years"]]
     references = [
         [["I", "am", "thirty", "six", "years", "old"], ["I", "am", "thirty", "six"]]
     ]
     bleu.compute(predictions=predictions, references=references)
```

```
[ ]  ! pip install sacrebleu
```

```
[ ]  sacrebleu = load_metric("sacrebleu")
     # SacreBLEU operates on raw text, not tokens
     predictions = ["I have thirty six years"]
     references = [["I am thirty six years old", "I am thirty six"]]
     sacrebleu.compute(predictions=predictions, references=references)
```

https://colab.research.google.com/github/huggingface/notebooks/blob/master/course/videos/bleu_metric.ipynb

https://huggingface.co/learn/nlp-course/chapter1/1

Performance of Transformer models- Good results

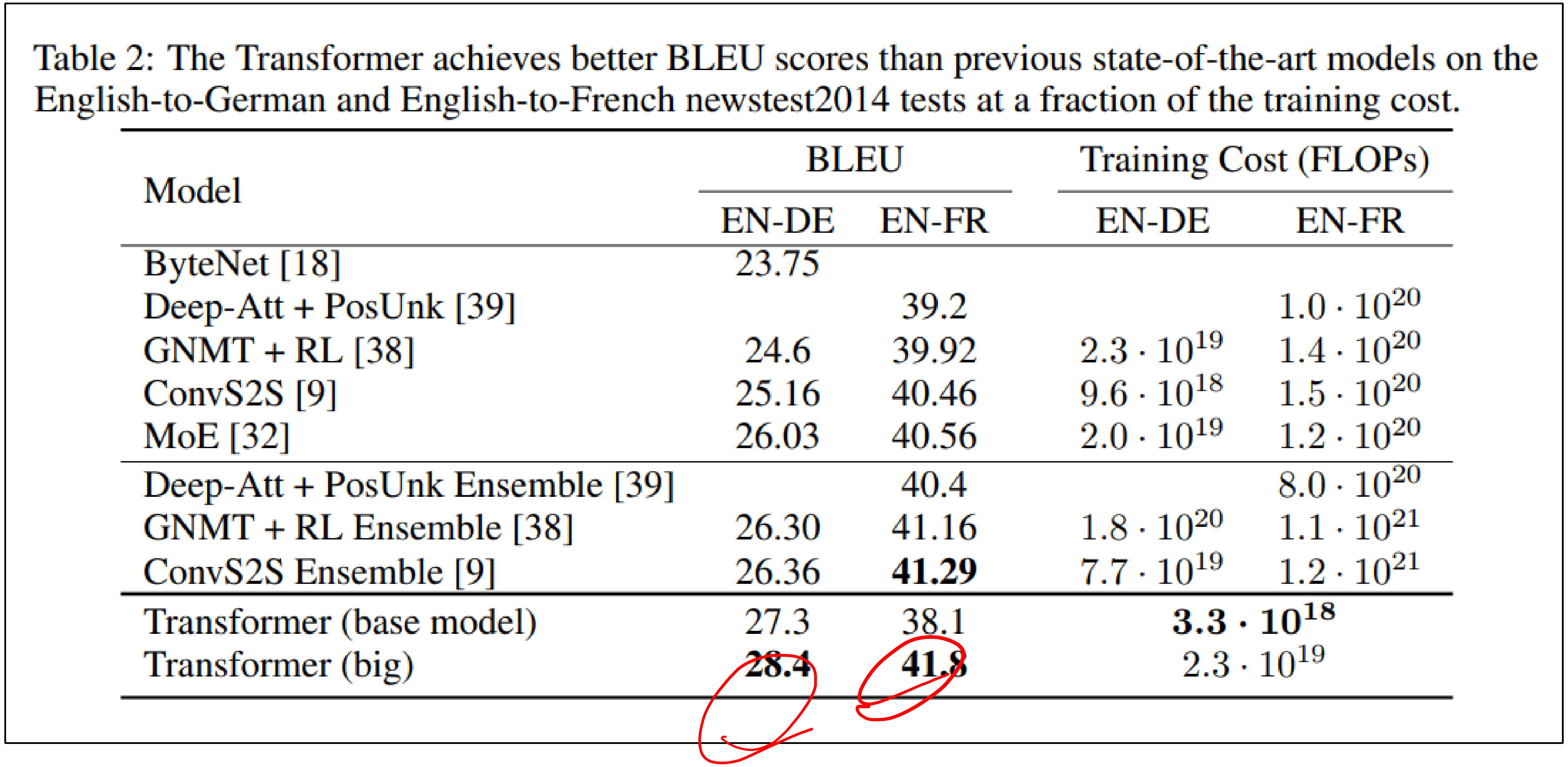
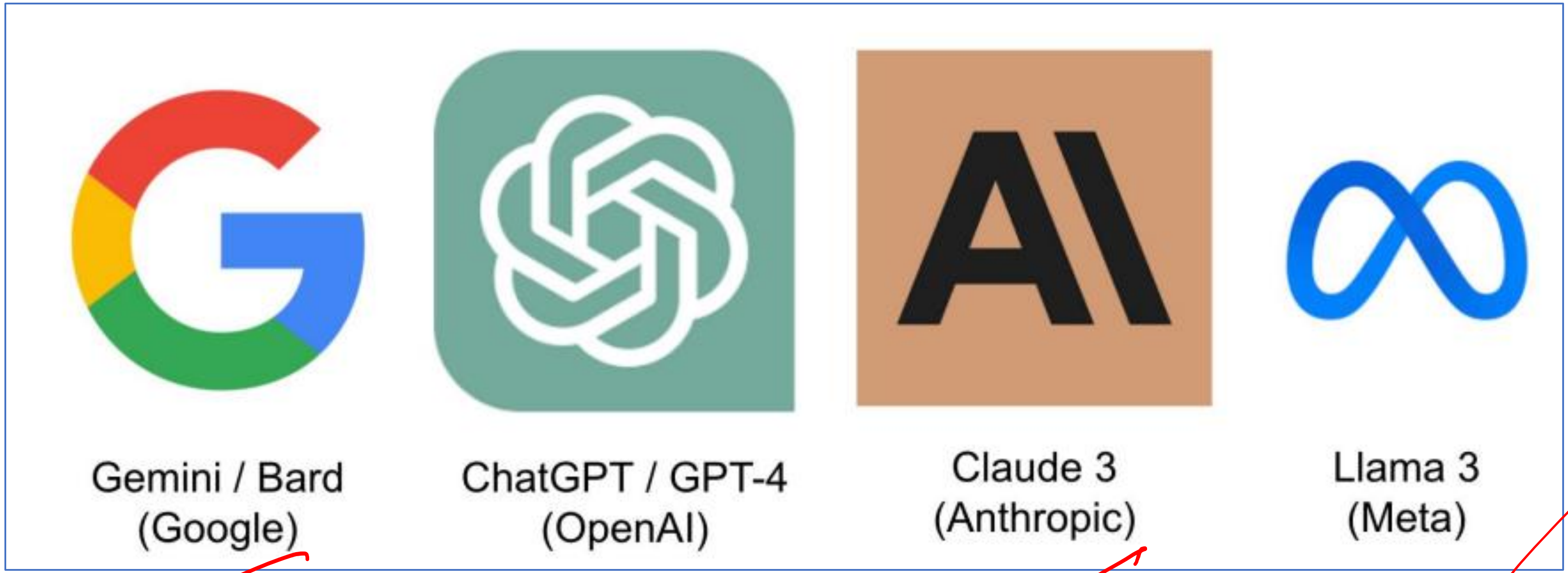


Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Recent Large Language Models



Gemini / Bard
(Google)

ChatGPT / GPT-4
(OpenAI)

Claude 3
(Anthropic)

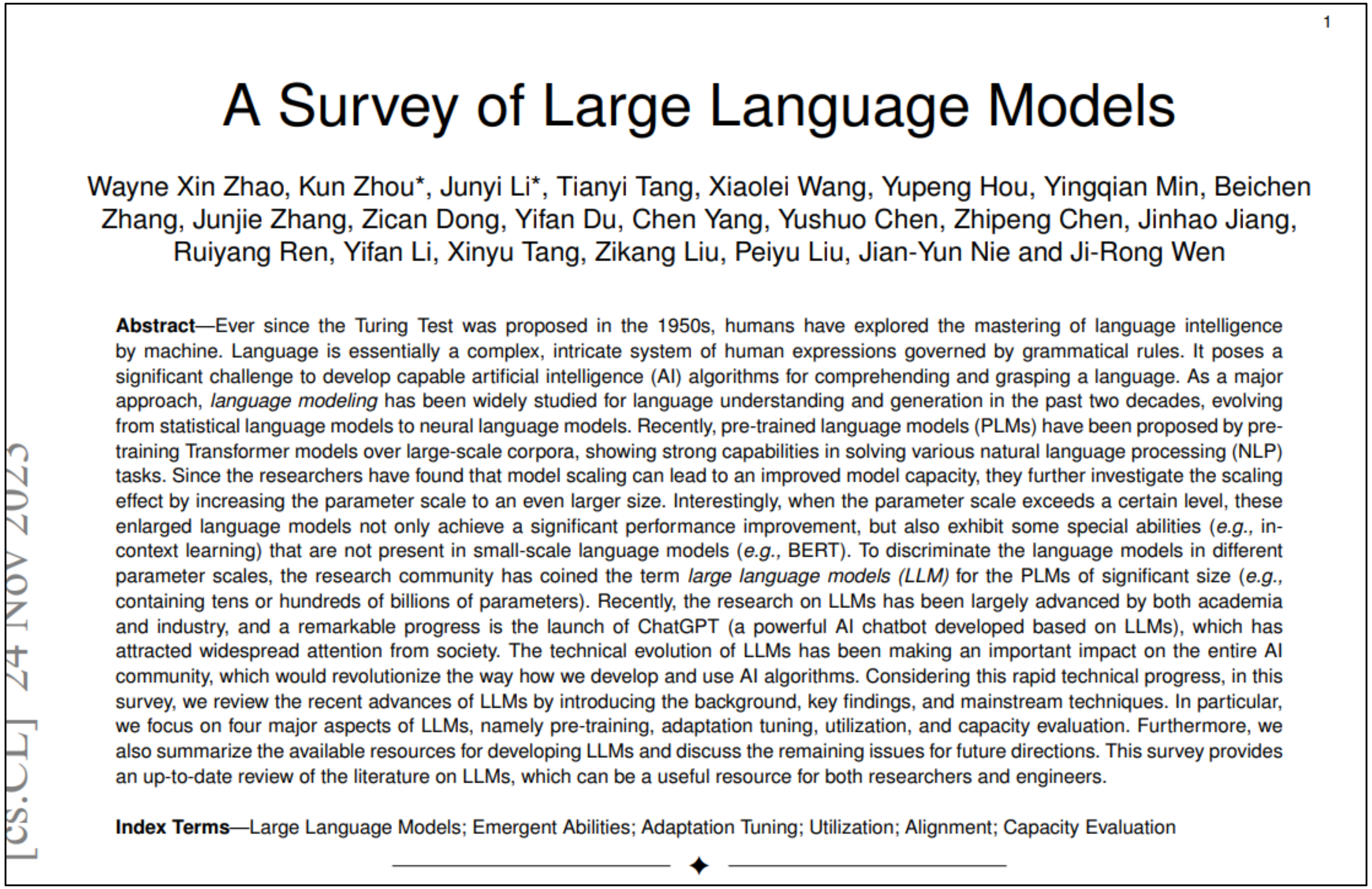
Llama 3
(Meta)

Leaderboard performance: SuperGlue

Super glue is the leaderboard over the range of challenging NLP tasks suchh as question answering, machine translation, CoRef, etc.

https://super.gluebenchmark.com/leaderboard/

Large Language models are on Hype!!!!!



1

A Survey of Large Language Models

Wayne Xin Zhao, Kun Zhou*, Junyi Li*, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie and Ji-Rong Wen

Abstract—Ever since the Turing Test was proposed in the 1950s, humans have explored the mastering of language intelligence by machine. Language is essentially a complex, intricate system of human expressions governed by grammatical rules. It poses a significant challenge to develop capable artificial intelligence (AI) algorithms for comprehending and grasping a language. As a major approach, *language modeling* has been widely studied for language understanding and generation in the past two decades, evolving from statistical language models to neural language models. Recently, pre-trained language models (PLMs) have been proposed by pre-training Transformer models over large-scale corpora, showing strong capabilities in solving various natural language processing (NLP) tasks. Since the researchers have found that model scaling can lead to an improved model capacity, they further investigate the scaling effect by increasing the parameter scale to an even larger size. Interestingly, when the parameter scale exceeds a certain level, these enlarged language models not only achieve a significant performance improvement, but also exhibit some special abilities (*e.g.,* in-context learning) that are not present in small-scale language models (*e.g.,* BERT). To discriminate the language models in different parameter scales, the research community has coined the term *large language models (LLM)* for the PLMs of significant size (*e.g.,* containing tens or hundreds of billions of parameters). Recently, the research on LLMs has been largely advanced by both academia and industry, and a remarkable progress is the launch of ChatGPT (a powerful AI chatbot developed based on LLMs), which has attracted widespread attention from society. The technical evolution of LLMs has been making an important impact on the entire AI community, which would revolutionize the way how we develop and use AI algorithms. Considering this rapid technical progress, in this survey, we review the recent advances of LLMs by introducing the background, key findings, and mainstream techniques. In particular, we focus on four major aspects of LLMs, namely pre-training, adaptation tuning, utilization, and capacity evaluation. Furthermore, we also summarize the available resources for developing LLMs and discuss the remaining issues for future directions. This survey provides an up-to-date review of the literature on LLMs, which can be a useful resource for both researchers and engineers.

Index Terms—Large Language Models; Emergent Abilities; Adaptation Tuning; Utilization; Alignment; Capacity Evaluation

https://arxiv.org/abs/2303.18223

What would you like to fix about Transformers?

Quadratic compute in self-attention (today):

- **Computing all pairs of interactions means our computation grows quadratically with the sequence length!**
- For recurrent models, it only grew linearly!
- **Position representations:**
- Are simple absolute indices the best we can do to represent position?
- As we learned: Relative linear position attention [Shaw et al., 2018]
- Dependency syntax-based position [Wang et al., 2019]
- Rotary Embeddings [Su et al., 2021]

We have come so far!! Final touches...

- 50+ hours of Assignment/Viva Evaluations
- 20+ hours of Project discussions and Evaluations
- Assignment-4 - Class activity
- Good luck for your End Sems!!!

Not the End.
Just the
Beginning.

Acknowledgments

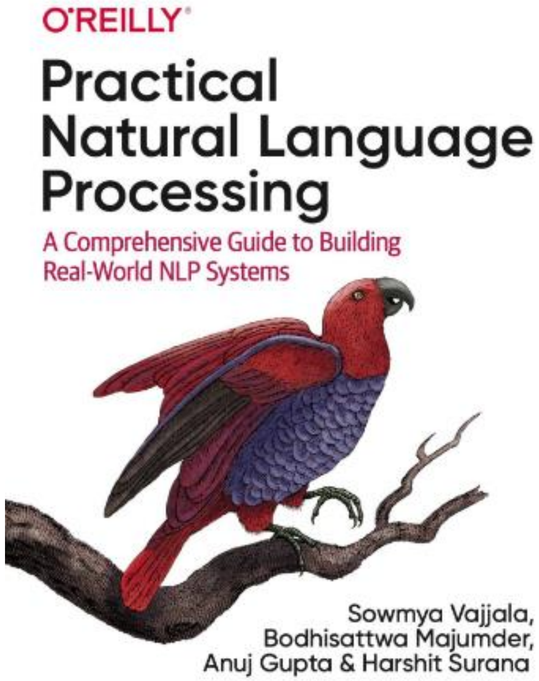
- These slides were adapted from the book SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
- Practical Natural Language Processing (A Comprehensive Guide to Building Real-World NLP Systems) O'reilly and some modifications from presentations and resources found in the WEB by several scholars.

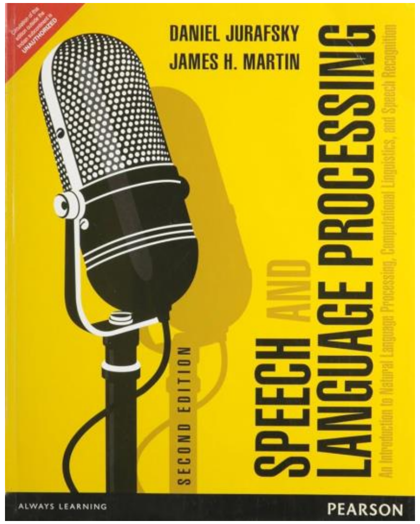
Reference materials

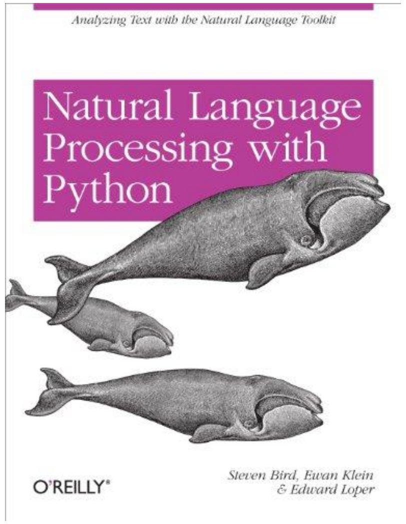
- https://vlanc-lab.github.io/mu-nlp-course/

- Lecture notes

- (A) Speech and Language Processing by Daniel Jurafsky and James H. Martin
- (B) Natural Language Processing with Python. (updated edition based on Python 3 and NLTK 3) Steven Bird et al. O’Reilly Media


O'REILLY
Practical Natural Language Processing
A Comprehensive Guide to Building Real-World NLP Systems
Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta & Harshit Surana


DANIEL JURAFSKY
JAMES H. MARTIN
SECOND EDITION
SPEECH AND LANGUAGE PROCESSING
An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
ALWAYS LEARNING
PEARSON


Analyzing Text with the Natural Language Toolkit
Natural Language Processing with Python
O'REILLY
Steven Bird, Ewan Klein & Edward Loper