A Brief History of Language Modeling

ASRU 2023 Tutorial, Taipei, Taiwan

Andreas Stolcke

uniphore

with credits and thanks to

SRI International Microsoft amazon

What is a language model?

- An estimator of the prior probability of a token sequence (e.g., text)
- Can be thought of as a sequential predictor:

$$P(w_1, w_2, \dots, w_n) = P(w_1)P(w_2|w_1) \ \dots \ P(w_n|\, w_1, \dots w_{n-1})$$

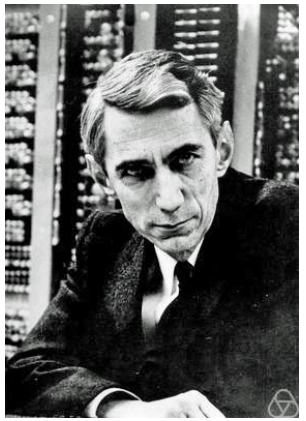
- Measures of goodness
  - Cross-entropy rate between a reference distribution (test set) and the model = average number of bits needed to encode a token

$$\frac{1}{n} H(w_1 \dots w_n) = -\frac{1}{n} \sum_i \log P(w_i|h_i)$$

  - Perplexity (average branching factor)

$$PPL = e^{\frac{1}{n} H(w_1 \dots w_n)}$$

Pre-History



- Claude Shannon (1916-2001)
  - Father of information theory
  - Inventor of language modeling: defined the “Shannon Game”
    - Let humans predict a text, one letter at a time
    - Record how many times the prediction is correct (or how many guesses are needed)
    - Derive an estimate for the “human entropy” of English: ≈ 1 bit / letter (*Bell System Technical Journal*, 1951)

Prediction and Entropy of Printed English

By C. E. SHANNON

(*ManuscriptReceived Sept. 15, 1950*)

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.

The Ages of Language Modeling

- Pre-history
- The age of N-grams (1980-2000)
- The age of structured LMs (1990-2010)
- The age of neural LMs (2000-?)
- The age of large LMs (2015-?)

The Age of N-grams

- Assumption: only nearby tokens will influence probability of next token
- Use statistics of N successive tokens (N-grams)
- Most popular for English word tokens: N = 3
- Popularized by the IBM speech group (Jelinek 1990)

Fred Jelinek (1932-2010)
"*Every time I fire a linguist our performance improves*"

SELF-ORGANIZED LANGUAGE MODELING FOR SPEECH RECOGNITION

by

F. Jelinek
Continuous Speech Recognition Group
IBM T.J.Watson Research Center
Yorktown Heights, N.Y. 10598

Trigram models and smoothing by interpolation

3. THE INTERPOLATED LANGUAGE MODEL AND ITS QUALITY

The language model of the current speech recognizer of the IBM Yorktown research group is based on a very simple equivalence classification: histories are equivalent if they end in the same two words. Thus

$$P(W) = \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) \tag{12}$$

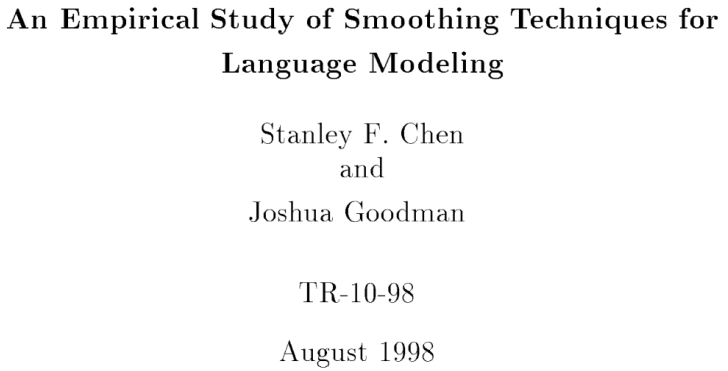
Originally we tried to estimate the basic trigram probabilities by the simple relative frequency approach

$$P(w_3 \mid w_1, w_2) = f(w_3 \mid w_1, w_2) = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)} \tag{13}$$

$$P(w_3 \mid w_1, w_2) = q_3 f(w_3 \mid w_1, w_2) + q_2 f(w_3 \mid w_2) + q_1 f(w_3) \tag{14}$$

N-gram Smoothing

- How do you estimate the probability of unseen N-grams?
- Many approaches
  - Good-Turing
  - Kneser-Ney
  - Absolute discounting
  - Interpolation
  - Interpolated Kneser-Ney
  - etc.
- Extensive comparison: Chen & Goodman (1998)


An Empirical Study of Smoothing Techniques for Language Modeling

Stanley F. Chen
and
Joshua Goodman

TR-10-98

August 1998

LM Training Objective (aka loss functions)

- Maximize likelihood of training data (ML training)
  - Minimizes training set perplexity
- Discriminative training
  - Optimize some decision based on LM
  - For example, N-best rescoring of ASR hypotheses
    - Max. posterior probability of correct hypothesis (maximum mutual information, MMI)
    - Min. word error of the top hypothesis (MWER training)
- Typical: ML on large general training set, discriminative on small target set

Beyond N-grams

- Problem with N-gram LMs:
  - Cannot use statistics that overlap
    - e.g., the number of adjacent bigrams together with the number of “skip-bigrams”
  - Ignore structure of linguistic dependencies

    The dog under the tree barked
    - *barked* is predicted by *dog*, it’s subject head, not by *tree*
  - Limited range, impossible to model other long-distance dependencies
    - Words tend to repeat
    - Topical coherence (“child” → “school”)
  - No generalization over words with similar properties

    My favorites pets are dogs|cats|hamsters

    Dogs|cats|hamsters needs a lot of care

N-gram models with “add-ons”

- Cache LM: boost recently seen tokens (Kuhn 1988)

A Cache-Based Natural Language Model for Speech Recognition

© Roland Kuhn, School of Computer Science

McGill University, Montreal

August, 1988.

- Mixture LMs: text coherence by interpolating multiple specialized LMs (e.g., Clarkson & Robinson, 1997)

LANGUAGE MODEL ADAPTATION USING MIXTURES AND AN EXPONENTIALLY DECAYING CACHE

P.R. Clarkson *A.J. Robinson*

Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, UK.
{prc14,ajr}@eng.cam.ac.uk

Class N-gram models

- Generalize n-grams to range over words and/or class labels

  My favorites pets are CLASS_PETS

  CLASS_PETS needs a lot of care

- Class labels "expand" to word tokens

  CLASS_PETS → *dogs* (0.4)|*cats* (0.2)|*hamsters* (0.05) | …

- Classes can be defined by hand (domain knowledge) or learned from training data by maximizing model likelihood (Brown et., 1992)

Class-Based *n*-gram Models of Natural Language

Peter F. Brown*
Peter V. deSouza*
Robert L. Mercer*
IBM T. J. Watson Research Center
Vincent J. Della Pietra*
Jenifer C. Lai*

Finite state graphs as language models

- Idea:
  - Walk through a weighted finite-state network (WFST) as tokens are read
    - Transition probabilities $P(s'|s, history)$
  - Next-token probabilities are a function of the state, or of state and history
    $P(w \mid s', history)$
- Hand-crafted state-based LMs are often used to encode domain knowledge

Computer Speech and Language (1996) **10,** 265–293

Stochastic automata for language modeling

Giuseppe Riccardi,† Roberto Pieraccini and Enrico Bocchieri

AT&T Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, U.S.A. email: dsp3/roberto/enrico@research.att.com

Finite-state LMs (continued)

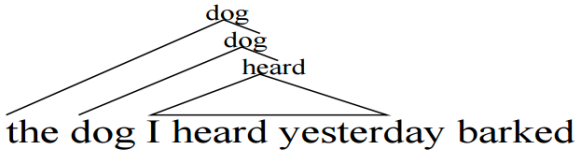
- N-gram LMs are a special case: previous $N-1$ words are the state
- Class N-grams are a special case: states correspond to classes
- LMs represented as FSTs have an algebra (intersection, composition)
  - E.g., compose the FST for class-ngrams with the FSTs for class membership
  - FSTs can be determinized for efficiency (deterministic = the next token determines the next state)

LMs for modeling linguistic structure

- N-grams ultimately are too limiting
- Build linguistic structure into the LM
  - Grammatical/semantic structure
  - Disfluencies (spontaneous speaking style)
  - Conversational structure

Grammatical Structure

- Predict words using the dependency structure of sentences
- Dependency tree is constructed incrementally by a statistical model



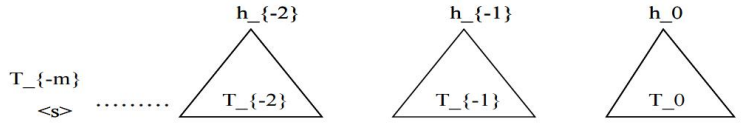


Figure 4: Before an adjoin operation

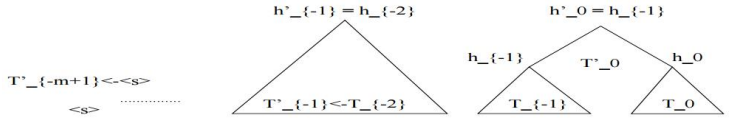


Figure 5: Result of adjoin-left

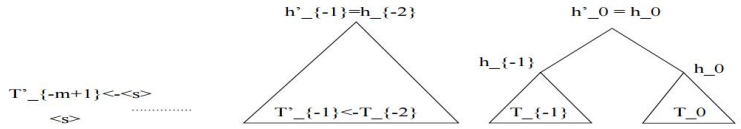


Figure 6: Result of adjoin-right

ELSEVIER

Computer Speech & Language

Volume 14, Issue 4, October 2000, Pages 283-332

Regular Article

Structured language modeling

Ciprian Chelba [a 1], Frederick Jelinek [b 2]

LM for Disfluencies: The Cleanup model

- Spontaneous language includes **disfluencies** due to online sentence planning: filled pauses (*uh*), repeated words, self-corrections, etc.
- Modeled with probabilistic insertion of disfluency events and editing of N-gram context to remove ("clean up") the extra words

I really don't uh <REPEAT> don't know what the big <REPEAT> the big deal is

Skip these subsequences when predicting the token after the <REPEAT> event

ICASSP 1996

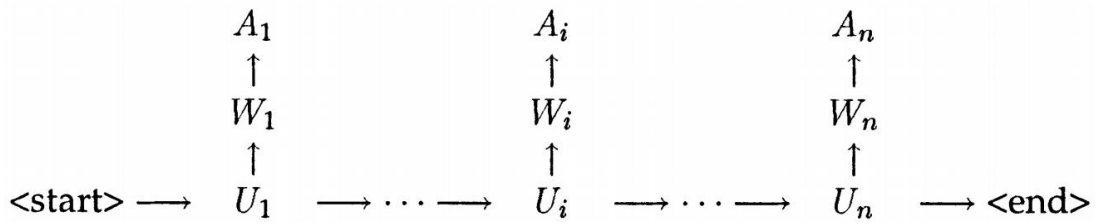
STATISTICAL LANGUAGE MODELING FOR SPEECH DISFLUENCIES

Andreas Stolcke *Elizabeth Shriberg*

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA 94025
stolcke@speech.sri.com
ees@speech.sri.com

Conversational Structure: Dialog Acts

- Conversations have a two-level structure:
  - Sequence of speaker labels and dialog acts: $P(S_2, U_2, |S_1, U_1, \dots)$
  - Sequences of words conditioned on the dialog acts $P(W_i|U_i)$



- Can also model acoustic and prosodic features A_i of dialog acts

Speaker	Dialogue Act	Utterance
A	YES-NO-QUESTION	So do you go to college right now?
A	ABANDONED	Are yo-,
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*it's my last year [laughter].*
A	DECLARATIVE-QUESTION	You're a, so you're a senior now.
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*I'm working on my projects trying to graduate [laughter].*
A	APPRECIATION	Oh, good for you.
B	BACKCHANNEL	*Yeah.*
A	APPRECIATION	That's great,
A	YES-NO-QUESTION	um, is, is N C University is that, uh, State,
B	STATEMENT	*N C State.*
A	SIGNAL-NON-UNDERSTANDING	What did you say?
B	STATEMENT	*N C State.*

Computational Linguistics (2000)

Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech

Andreas Stolcke*
SRI International

Klaus Ries
Carnegie Mellon University and University of Karlsruhe

Noah Coccaro
University of Colorado at Boulder

Elizabeth Shriberg
SRI International

Rebecca Bates
University of Washington

Daniel Jurafsky
University of Colorado at Boulder

Paul Taylor
University of Edinburgh

Rachel Martin
Johns Hopkins University

Carol Van Ess-Dykema
U.S. Department of Defense

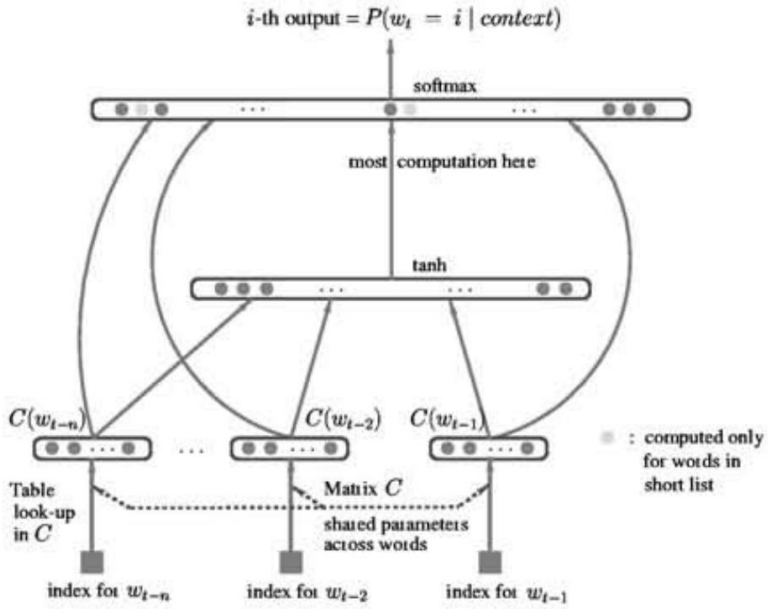
Marie Meteer
BBN Technologies

The Age of Neural LMs

- Three distinct phases
  - Feed-forward neural N-gram models
  - Recurrent neural models
  - Transformer-based models (and dramatic scaling)
  - Instruction-tuned models, in-context learning

Feedforward Neural LMs

- Combine three key ideas
  - Encode N-gram context by (N-1)-tuples of word embeddings
  - Use deep neural classifiers to predict next word token using softmax
  - Learn word embeddings jointly with the word prediction task

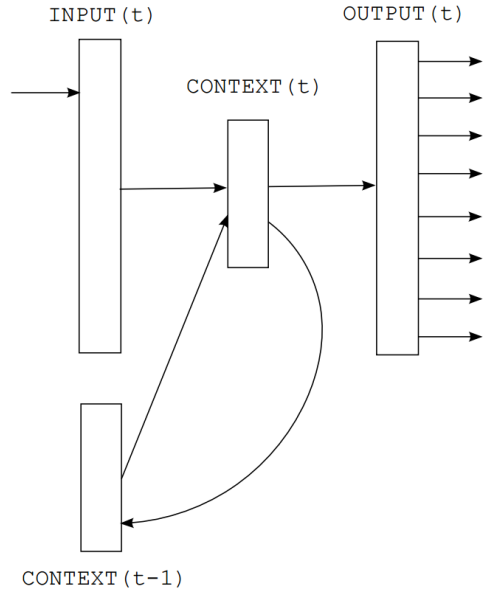


NIPS 2000

A Neural Probabilistic Language Model

Yoshua Bengio, Réjean Ducharme and Pascal Vincent
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{bengioy,ducharme,vincentp}@iro.umontreal.ca

Recurrent Neural Nets (RNNs)



- Context vectors: feed hidden layer activation back into next-state and next-word prediction
- Allows history memory beyond fixed window
  - Continuous-space version of FST
- Later enhanced by
  - Long Short-Term Memory gating
  - Stacked recurrent layers

INTERSPEECH 2010

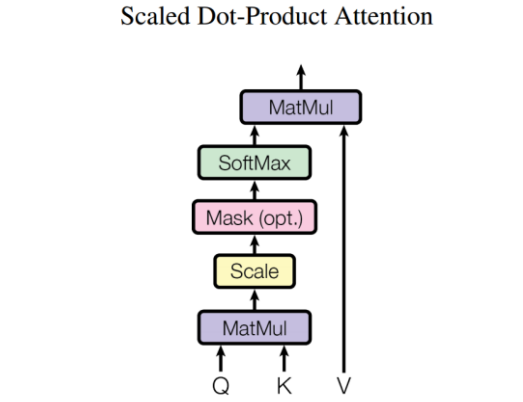
Recurrent neural network based language model

Tomáš Mikolov[1,2], *Martin Karafiát*[1], *Lukáš Burget*[1], *Jan "Honza" Černocký*[1], *Sanjeev Khudanpur*[2]

[1]Speech@FIT, Brno University of Technology, Czech Republic
[2] Department of Electrical and Computer Engineering, Johns Hopkins University, USA
{imikolov,karafiat,burget,cernocky}@fit.vutbr.cz, khudanpur@jhu.edu

(Neural) Transformers

- A new NN architecture designed to learn long-range dependencies
- Based on query-key-value self-attention at multiple layers of representation



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

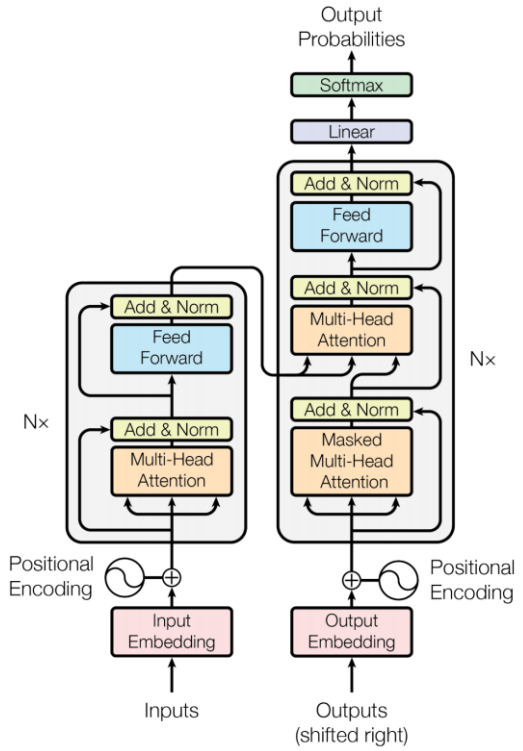


Figure 1: The Transformer - model architecture.

NIPS 2017

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

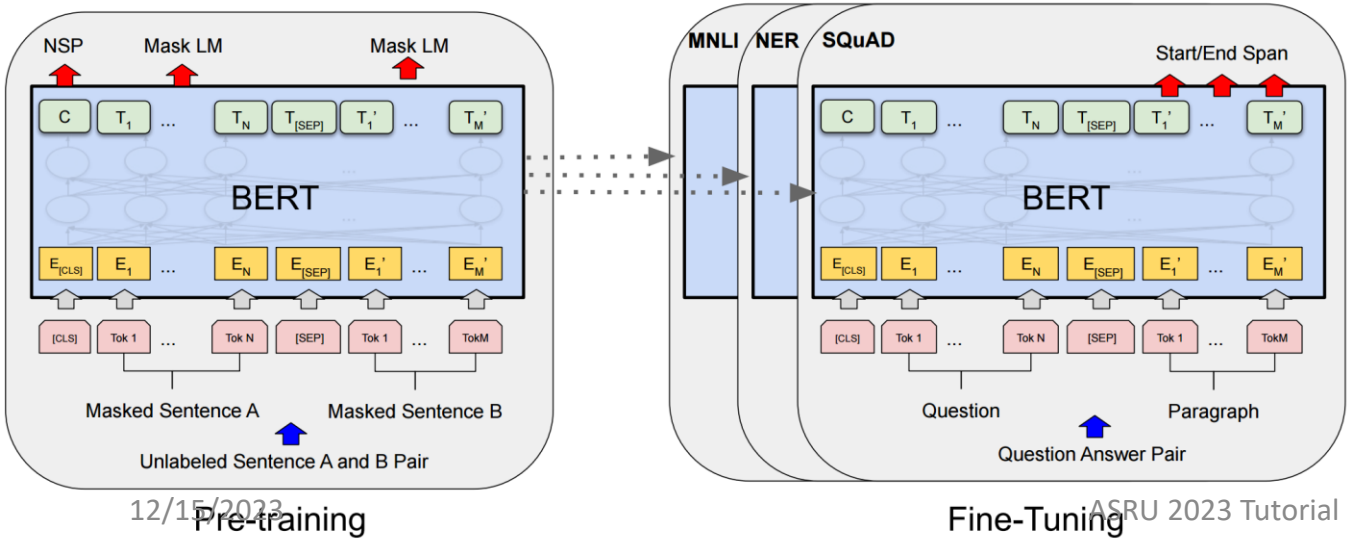
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

BERT - **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

- Pre-trained by
  - Predicting to randomly masked portions of the input sequence
  - Next-sentence prediction
- Learns contextual embeddings for input tokens and entire sequences
- Fine-tuned on a variety of NLP tasks, from [CLS] embedding



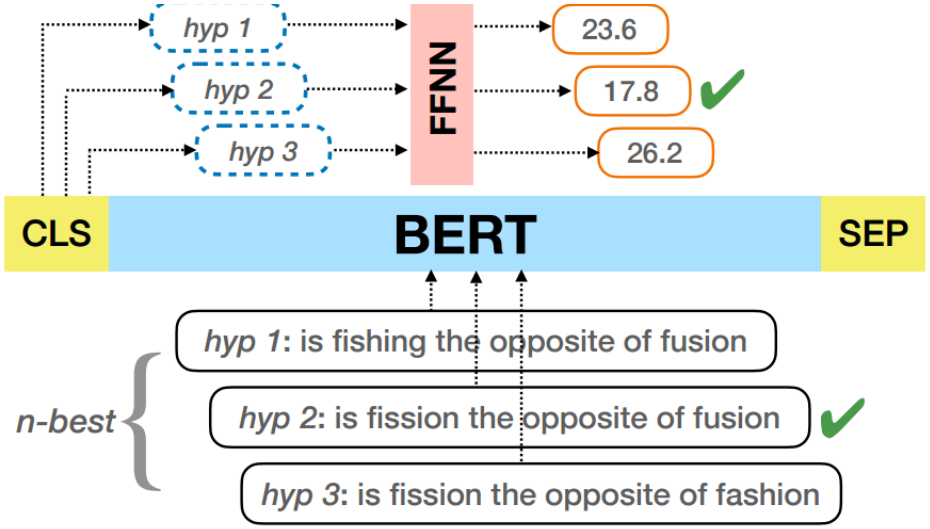
arXiv 2018

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

RescoreBERT – Discriminative LM from BERT

- Fine-tune BERT with two objectives
  - regress on (distill) a masked LM pseudo-likelihood from the CLS embedding
  - minimize word error over an N-best list (discriminative LM)



ICASSP 2022

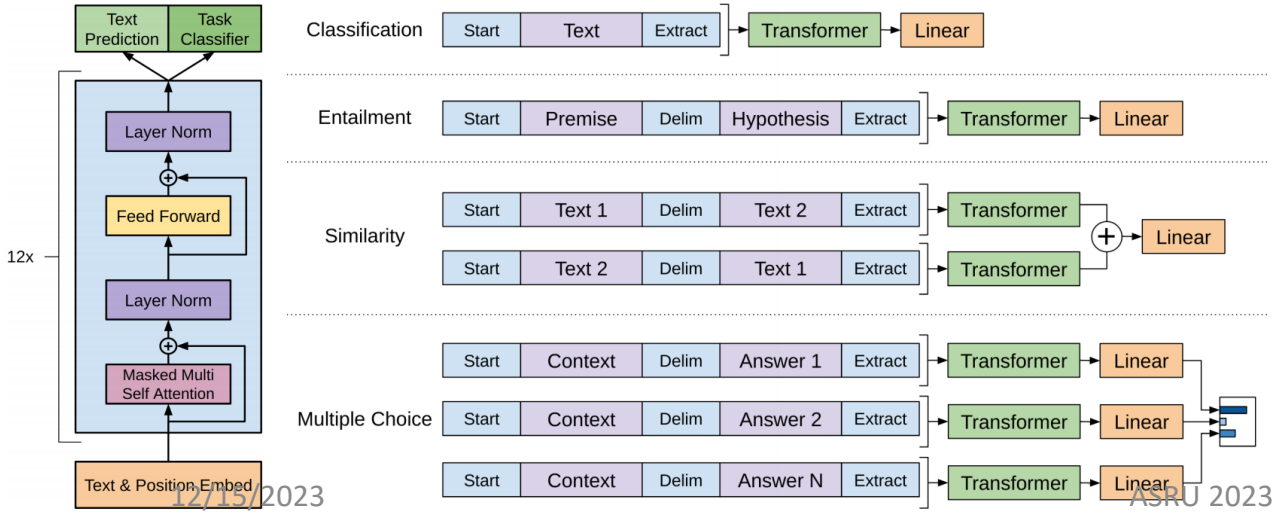
RESCOREBERT: DISCRIMINATIVE SPEECH RECOGNITION RESCORING WITH BERT

Liyan Xu[1,2] *Yile Gu*[1] *Jari Kolehmainen*[1] *Haidar Khan*[1] *Ankur Gandhe*[1]
Ariya Rastrow[1] *Andreas Stolcke*[1] *Ivan Bulyko*[1]

[1]Amazon Alexa AI, USA [2]Emory University, USA

GPT – Generative Pre-Training

- Transformer based
- Pre-train on token prediction task (left-to-right, unlike BERT)
- Supervised fine-tuning on a specific NL understanding task
  - inference, question answering, semantic similarity, text classification
  - Regularize with token prediction loss
  - Based on final token hidden embedding



OpenAI 2018

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

Comparative Study of LLMs as LMs for ASR

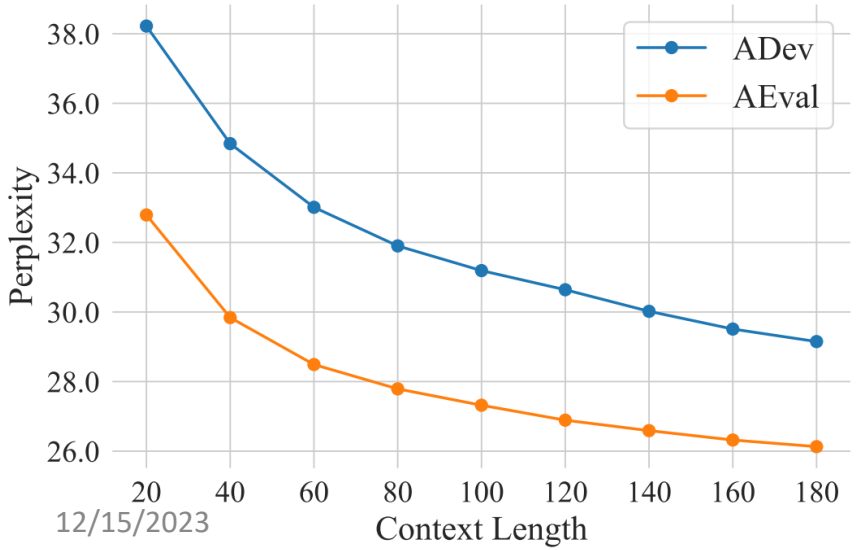
- Some conclusions:
  - Even with only small in-domain pre-training,

    Transformer LMs > other NN LMs
  - Fine-tuning LLMs on in-domain LM task helps
  - Context beyond current sentence (utterance) helps
  - GPT and BERT are complementary

Model	ADev	AEval	SWB	CH
4-gram	19.9	20.2	8.6	17.0
FNN LM	19.4	19.5	7.9	15.8
LSTM LM	18.2	17.9	6.7	13.7
Transformer LM	18.4	18.4	6.6	13.7
F ⊕ L ⊕ T	**17.9**	**17.7**	**6.5**	**13.5**

Table 1. %WER on AMI (ADev and AEval) and on *eval2000*

GPT	×	19.2	18.9	-	-
	√	16.5	16.0	6.3	13.1
GPT ⊕ GPT-2	√	**16.0**	**15.6**	**6.1**	**12.7**
GPT ⊕ GPT-2 ⊕ BERT		**15.9**	**15.5**		



ASRU 2021

ADAPTING GPT, GPT-2 AND BERT LANGUAGE MODELS FOR SPEECH RECOGNITION

Xianrui Zheng, Chao Zhang, Philip C. Woodland

Cambridge University Engineering Dept., Trumpington St., Cambridge, CB2 1PZ U.K.

{xz396, cz277, pcw}@eng.cam.ac.uk

Prompting LLMs

- LLMs can be fine-tuned to interpret instructions (cf. ChatGPT)
  → Prompt engineering
  → In-context learning (instructing with examples)
- It is possible to prompt LLMs to perform evaluation, ranking etc. of input texts for specific purposes
- For ASR rescoring and/or correction: see Huck's paper & presentation

ASRU 2023

GENERATIVE SPEECH RECOGNITION ERROR CORRECTION WITH LARGE LANGUAGE MODELS AND TASK-ACTIVATING PROMPTING

Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, Andreas Stolcke

Amazon, USA

What has changed, and What has stayed the same

- Hand-crafted dependency structures in models have been replaced with large “dumb” neural nets that can learn the dependencies
- But they need the right architectural features to learn the dependencies (cf. self-attention mechanism)
- It’s still good to understand linguistic phenomena so that we know what aspects of the data to encode (context, speaker info) for learning
- Training on large mismatched data, then fine-tuning (or interpolating) with in-domain data is still good.
- Ensembling of different architectures and/or training data sets still a good idea

Thank you!

andreas.stolcke@uniphore.com