Masked Language Models

BERT

Masked Language Modeling

- We've seen autoregressive (causal, left-to-right) LMs.
- But what about tasks for which we want to peak at future tokens?
	- Especially true for tasks where we map each input token to an output token
- **Bidirectional** encoders use **masked self-attention** to
	- map sequences of input embeddings (x1,...,x*n*)
	- to sequences of output embeddings of the same length (h1,...,h*n*),
	- where the output vectors have been contextualized using information from the entire input sequence.

Bidirectional Self-Attention

a) A causal self-attention layer b) A bidirectional self-attention layer

q4•k1 q4•k2 q4•k3 q4•k4 q3•k1 q3•k2 q3•k3 q1•k2 q1•k3 q1•k4 q2•k3 q2•k4 q3•k4

 $\overline{\sqrt{d_k}}$

N

elf-attention

BERT: Bidirectional Encoder Representations from Transformers

BERT (Devlin et al., 2019)

- 30,000 English-only tokens (WordPiece tokenizer)
- Input context window *N*=512 tokens, and model dimensionality *d*=768
- *^L*=12 layers of transformer blocks, each with *A*=12 (bidirectional) multihead- attention layers.
- The resulting model has about 100M parameters.

- 250,000 multilingual tokens (SentencePiece Unigram LM tokenizer)
- Input context window *N*=512 tokens,model dimensionality *d*=1024
- *L*=24 layers of transformer blocks, with *A*=16 multihead attention layers each
- The resulting model has about 550M parameters.

XLM-RoBERTa (Conneau et al., 2020)

Masked Language Models

BERT

Masked Language¹ Models

Masked LM training

Masked training intuition

• For **left-to-right LMs**, the model tries to predict the last word from prior words:

The water of Walden Pond is so beautifully $\sqrt{ }$

The **contract of Walden Pond So beautifully blue** • The model generates a probability distribution over the vocabulary for each

- And we train it to improve its predictions.
- For **bidirectional masked LMs**, the model tries to predict one or more words from all the rest of the words:

- missing token
- We use the cross-entropy loss from each of the model's predictions to drive the learning process.

MLM training in BERT

15% of the tokens are randomly chosen to be part of the masking Example: "Lunch was **delicious**", if delicious was randomly chosen: Three possibilities:

- 1. 80%: Token is replaced with special token [MASK] Lunch was **delicious ->** Lunch was **[MASK]**
- 2. 10%: Token is replaced with a random token (sampled from unigram prob) Lunch was **delicious ->** Lunch was **gasp**
- 3. 10%: Token is unchanged

Lunch was **delicious ->** Lunch was **delicious**

In detail

$$
\mathbf{u}_i = \mathbf{h}_i^{\text{L}} \mathbf{E}^{\text{T}}
$$

$$
\mathbf{y}_i = \text{softmax}(\mathbf{u}_i)
$$

$$
L_{MLM}(x_i) = -\log P(x_i | \mathbf{h}_i^L)
$$

 \blacksquare masked be *M*, the version of that sentence with some tokens replaced by masks be The gradients by taking the average of this loss over the batch updates are based versions were parameter Γ \blacksquare We get the gradients by taking the average of this loss over the batch

$$
L_{MLM} = -\frac{1}{|M|} \sum_{i \in M} \log P(x_i | \mathbf{h}_i^L)
$$

MLM loss 6 CHAPTER 11 • MASKED LANGUAGE MODELS With a predicted probability distribution for each masked item, we can use crossentropy to compute the loss for each masked item—the negative log probability in the negative log probability of \mathbb{R}^n

sequences).

assigned to the set of the vocabulary of the vocabulary of final transformer layer L, multiplies it by a The LM head takes output of final transformer layer L, multiplies it by given vector of input to interesting to the sense of the sense of the sense of the sense of the set of top set of the sense of the sense of the sense of the sense of tokens that are $\frac{1}{2}$ are $\frac{1}{2}$ are $\frac{1}{2}$ a masked be *M*, the version of that sentence with some tokens replaced by masks be **1** The LM head takes output of final transformer layer L, multiplies it by
Inperchading layer and turns into probabilities: unembedding layer and turns into probabilities:

entropy to compute the loss for each masked item—the negative log probability

assigned to the actual masked word, as shown in Fig. 11.3. More for a shown in Fig. 11.3. More formally, for a
As shown in Fig. 11.3. More formally, formally, formally, formally, for a shown in Fig. 11.3. More formally, f

x word *long*, given output h^L ;): the word **long** in Fig. 1.3, the word *long* in Fig. 2, the loss is the probability of the correct word **long**, given by $\frac{1}{2}$ *E.g., for the x_i corresponding to "long", the loss is the probability of the correct word <i>long* given output b^L 1[.]

Next Sentence Prediction

- [CLS] is prepended to the input sentence pair, I cap is prepended to the imput sentence pair,
- [SEP] is placed between the sentences, and also after second sentence And two more special tokens
- $[1st segment]$ and $[2nd segment]$ \bullet [1st segment] and $[2^{nd}$ segment]
- These are added to the input embedding and positional embedding we add a special head, in this case and a special head, which consists of a learned set of a learned set of a l
In this case and a learned set of the annu classification weights was a the decomposition weights was a two-class produced was the produced from the class produced from

BERT introduces two special tokens

h^L_{CLS} from the final layer [CLS] token is input to classifier head (weights W_{NSP}) that products two classes: predicts two classes:.

Given 2 sentences the model predicts if they are a real pair of adjacent sentences from the training corpus or a pair of unrelated sentences. 11.2 • TRAINING BIDIRECTIONAL ENCODERS 7

$$
\mathbf{y}_i = \text{softmax}(\mathbf{h}_{\text{CLS}}^L \mathbf{W}_{\text{NSP}})
$$

NSP Loss with classification head

More details

- Original model was trained with 40 passes over training data Some models (like RoBERTa) drop NSP loss
- Tokenizer for multilingual models is trained from stratified sample of languages (some data from each language)
- Multilingual models are better than monolingual models with small numbers of languages
- With large numbers of languages, monolingual models in that language can be better
- The "curse of multilinguality"

Masked Language¹ Models

Masked LM training

Masked Language Models

Contextual Embeddings

Contextual Embeddings to represent words

Static vs Contextual Embeddings

Static embeddings represent **word types** (dictionary entries) Contextual embeddings represent **word instances** (one for each time the word occurs in any context/sentence)

Players must always move a token according to the **die** value

Contextual embeddings offer a continuous high-dimensional model of meaning that is more fine grained than discrete senses. also be visualized geometrically. Fig. 11.6 shows a two-dimensional projection of many instances of the BERT embeddings of the BERT embeddings of the William and German and German and German.
Extragal conditions of the word of the word of the German and German and German and German and German and Germ earling that is more line gramed than discrete senses.

Word sense polysemous (from Greek 'many senses', *poly-* 'many' + *sema*, 'sign, mark').²

Words are ambiguous s are ambiguous mouse1 and mouse2. These senses can be found listed in online thesauruses (or

10 CHAPTER 11 • MASKED LANGUAGE MODELS IN 1999 POST AND THE CHAPTER 11 • MASKED LANGUAGE MODELS IN 1999 POST A
The contract of the chapter in the contract of the chapter in the contract of the contract of the contract of

A word sense is a discrete representation of one aspect of meaning

mouse1 : a *mouse* controlling a computer system in 1968. mouse2 : a quiet animal like a *mouse* **bank**¹ : ...a *bank* can hold the investments in a custodial account ... bank² : ...as agriculture burgeons on the east *bank*, the river ...

Word sense disambiguation (WSD)

The task of selecting the correct sense for a word.

1-nearest neighbor algorithm for WSD \overline{a} t-nearest neighbor algorithm for wab 1-nearest neighbor algorithm for WSD

Melamud et al (2016), Peters et al (2018) $\mathcal{L}(\mathcal{L})$ is common to pool multiple layers by summing the vector representations $\mathcal{L}(\mathcal{L})$

At training time, take a sense-labeled corpus like SEMCOR Run corpus through BERT to get contextual embedding for each token (There are various ways to compute this contextual embedding *vi* for a token *i*; for μ training three, take a sense-labeled corpus like suivicon At training time, take a sense-labeled corpus like SEMCOR for each token the *numb* BERT to get contextual embedding for each token

• E.g., pooling representations from last 4 BERT transformer layer Then for each sense s of word w for n tokens of that sense, pool embeddings: *vi* victor of *view victor sense contextual set* 1 $\sqrt{ }$ *vi* to produce a contextual sense extending variations from last 1 DE v*^s* = <u>.</u>
F vor a w for if tokens of that sense, pool

At test time, given a token of a target word *t*, compute contextual embedding t and choose its nearest neighbor sense from training set α test time, given a token of a target word c, compute contextual embed.emig t and choose its nearest neighbor sense from the training set, i.e., $\frac{1}{2}$ At test time, given a token of a target word t , compute contextual \mathcal{L} embedding t and choose its hearest neighbor sense

$$
\mathbf{v}_s = \frac{1}{n} \sum_i \mathbf{v}_i \qquad \forall \mathbf{v}_i \in \text{ tokens}(s)
$$

$$
sense(t) = \underset{s \in senses(t)}{\operatorname{argmax}} \ cosine(t, v_s)
$$

1-nearest neighbor algorithm for WSD

Similarity and contextual embeddings

- We generally use cosine as for static embeddings
- But some issues:
	- Contextual embeddings tend to be **anisotropic:** all point in roughly the same direction so have high inherent cosines (Ethayarajh 2019)
	- Cosine measure are dominated by a small number of "rogue" dimensions with very high values (Timkey and van Schijndel 2021)
	- Cosine tends to underestimate human judgments on similarity of word meaning for very frequent words (Zhou et al., 2022)

Masked Language Models

Contextual Embeddings

Masked Language Models

Fine -Tuning for Classificatio n

Adding a sentiment classification head

Sequence-Pair classification

Assign a label to pairs of sentences:

- paraphrase detection (are the two sentences paraphrases of each other?)
- logical entailment (does sentence A logically entail sentence B?)
- discourse coherence (how coherent is sentence B as a follow-on to sentence A?)

Pairs of sentences are given one of 3 labels the meaning of the sentence (the hypothesis). Here are representative examples are given one or β

Example: Natural Language Inference inference entails a model is presented with a pair of sentences and must classify the re- σ : Natural Language Interence sentences are given one of 3 labels: *entails*, *contradicts* and *neutral*. These labels

Algorithm: pass the premise/hypothesis pairs through a bidirectional encoder and use the output vector for the [CLS] token as the input to the classification head . *tails* means that the premise entails the hypothesis; *neutral* means that neither is

- Neutral
	- a: Jon walked back to the town to the smithy.
	- b: Jon traveled back to his hometown.
- Contradicts
	- a: Tourist Information offices can be very helpful.
	- b: Tourist Information offices are never of any help.
- Entails
	- a: I'm confused.
	- b: Not all of it is very clear to me.

Fine-tuning for sequence labeling

Assign a label from a small fixed set of labels to each token in the sequence.

- Named entity recognition
- Part of speech tagging

.

Named Entity Recognition city necognition

A named entity is anything that can be referred to with a proper name: a person, a location, an organization $\mathcal{P}(P|P|P) = \mathcal{P}(P|P|P)$ said. $\mathcal{P}(P|P|P) = \mathcal{P}(P|P|P)$ said. $\mathcal{P}(P|P|P) = \mathcal{P}(P|P|P)$ satisfies $\mathcal{P}(P|P|P) = \mathcal{P}(P|P|P)$ t , and α is α and α α α is α is α .

Named entity recognition (NER): find spans of text that constitute proper names and tag the type of the entity

Puter science. the cyclone. ne Canyon. es for parking.

Named Entity Recognition

16 CHAPTER 11 • MASKED LANGUAGE MODELS IN 16 ° MASKED LANGUAGE MODELS IN 16 ° MASKED LANGUAGE MODELS IN 16 ° MA
16 SEPTEMBER 11 • MASKED LANGUAGE MODELS IN 16 ° MASKED LANGUAGE MODELS IN 16 ° MASKED LANGUAGE MODELS IN 16 °

Citing high fuel prices, $[**ORG**$ United Airlines] said $[**TIME**$ Friday] it has increased fares by $\left[\text{MONEY} \right]$ for round trip on flights to some cities also served by lower-cost carriers. $[ORG]$ **American Airlines**, a unit of $[ORG$ **AMR Corp.**], immediately matched the move, spokesman [$PIER$ Tim Wagner] said. [ORG United], a unit of $[ORG$ UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as $\left[\begin{array}{cc} R_1 \end{array}\right]$ to L_{LOC} Dallas] and L_{LOC} Denver] to L_{LOC} San Francisco].

Ramshaw and Marcus (1995)

A method that lets us turn a segmentation task (finding boundaries of entities) into a classification task the boundary and the named entity type. Consider the following sentence:

> [PER Jane Villanueva] of [_{ORG} United], a unit of [_{ORG} United Airlines Holding], said the fare applies to the $\left[$ _{LOC} Chicago</sub> $\right]$ route. t represent the same information of the same in the same in the same in the bracket of the bracket notation, but has the bracket of the advantage that we can represent the same simple sequence model the same simple sequence model in the same simple seq

used to generate the final output tag sequence. Fig. 11.13 illustrates an example of

sequence labeling
this approach, where $\frac{1}{2}$ is a vector of probabilities of probabilities over the tags. And *k* is a vector of probabilities of probabilities of probabilities of probabilities of probabilities of prob

$$
y_i = softmax(h_i^L W_K)
$$

$$
t_i = argmax_k(y_i)
$$

More details

We need to map between tokens (used by LLM) and words (used in definition of name entities)

We evaluate NER with F1 (precision/recall)

Masked Language Models

Fine -Tuning for Classificatio n