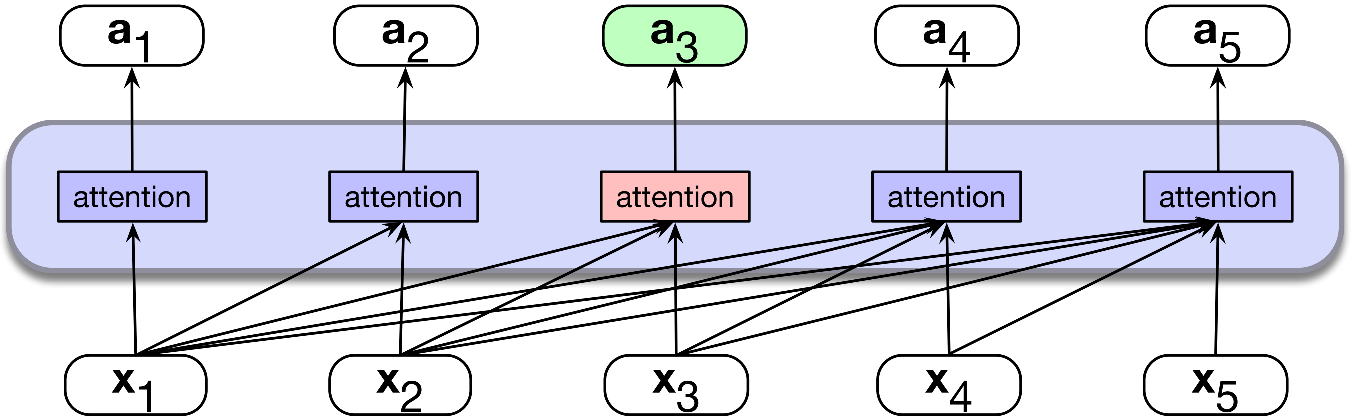
Masked Language Models

BERT

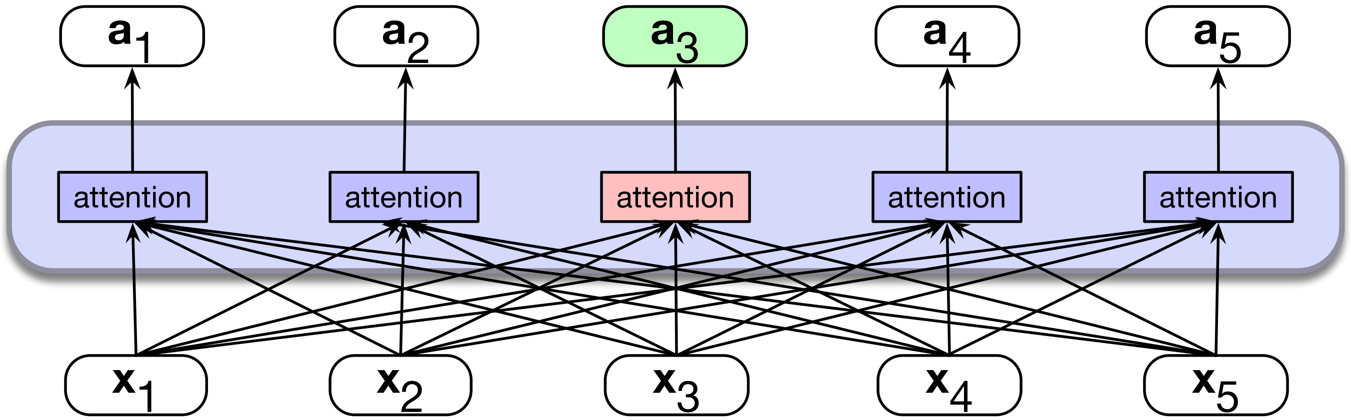
Masked Language Modeling

- 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 ($x1,...,xn$)
  - to sequences of output embeddings of the same length ($h1,...,hn$),
  - 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

Easy! We just remove the mask

Casual self-attention

N

q1•k1	$-\infty$	$-\infty$	$-\infty$
q2•k1	q2•k2	$-\infty$	$-\infty$
q3•k1	q3•k2	q3•k3	$-\infty$
q4•k1	q4•k2	q4•k3	q4•k4

N

$$\mathbf{head} = \mathrm{softmax}\left(\mathrm{mask}\left(\frac{\mathbf{QK}^{\mathsf{T}}}{\sqrt{d_k}}\right)\right)\mathbf{V}$$

Bidirectional self-attention

N

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

N

$$\mathbf{head} = \mathrm{softmax}\left(\frac{\mathbf{QK}^{\mathsf{T}}}{\sqrt{d_k}}\right)\mathbf{V}$$

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.

XLM-RoBERTa (Conneau et al., 2020)

- 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.

Masked Language Models

BERT

Masked Language 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 ______

  - 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:

  The ________ of Walden Pond ________ so beautifully blue

  - The model generates a probability distribution over the vocabulary for each 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]**

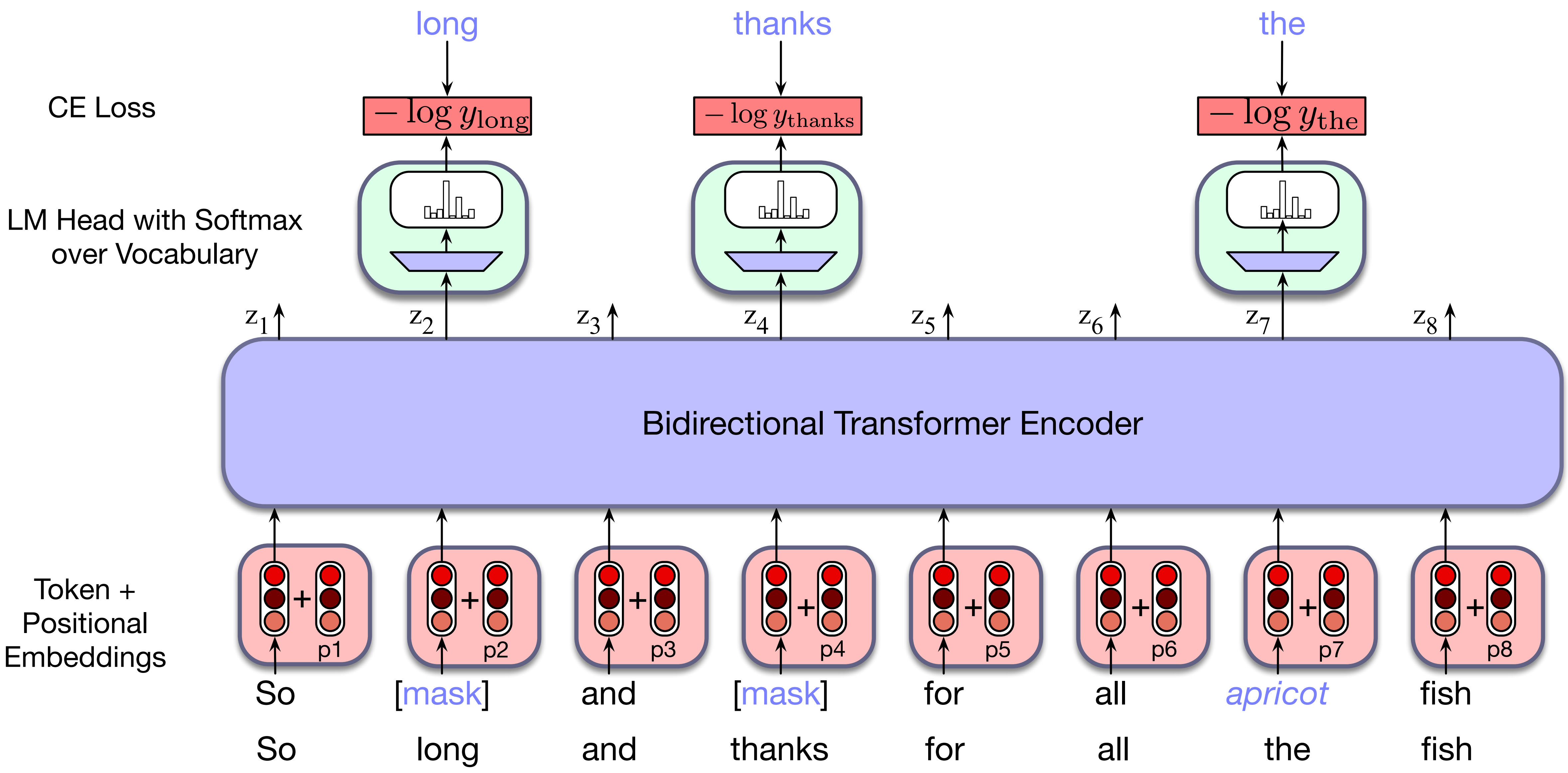
2. 10%: Token is replaced with a random token (sampled from unigram prob)

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

3. 10%: Token is unchanged

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

In detail


long
thanks
the
CE Loss
$-\log y_{\mathrm{long}}$
$-\log y_{\mathrm{thanks}}$
$-\log y_{\mathrm{the}}$
LM Head with Softmax over Vocabulary
z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8
Bidirectional Transformer Encoder
Token + Positional Embeddings
p1 p2 p3 p4 p5 p6 p7 p8
So [mask] and [mask] for all apricot fish
So long and thanks for all the fish

MLM loss

The LM head takes output of final transformer layer L, multiplies it by unembedding layer and turns into probabilities:

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

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

E.g., for the x_i corresponding to "long", the loss is the probability of the correct word *long*, given output h^L_i):

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

We get the gradients by taking the average of this loss over the batch

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

Next Sentence Prediction

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.

BERT introduces two special tokens

- [CLS] is prepended to the input sentence pair,
- [SEP] is placed between the sentences, and also after second sentence

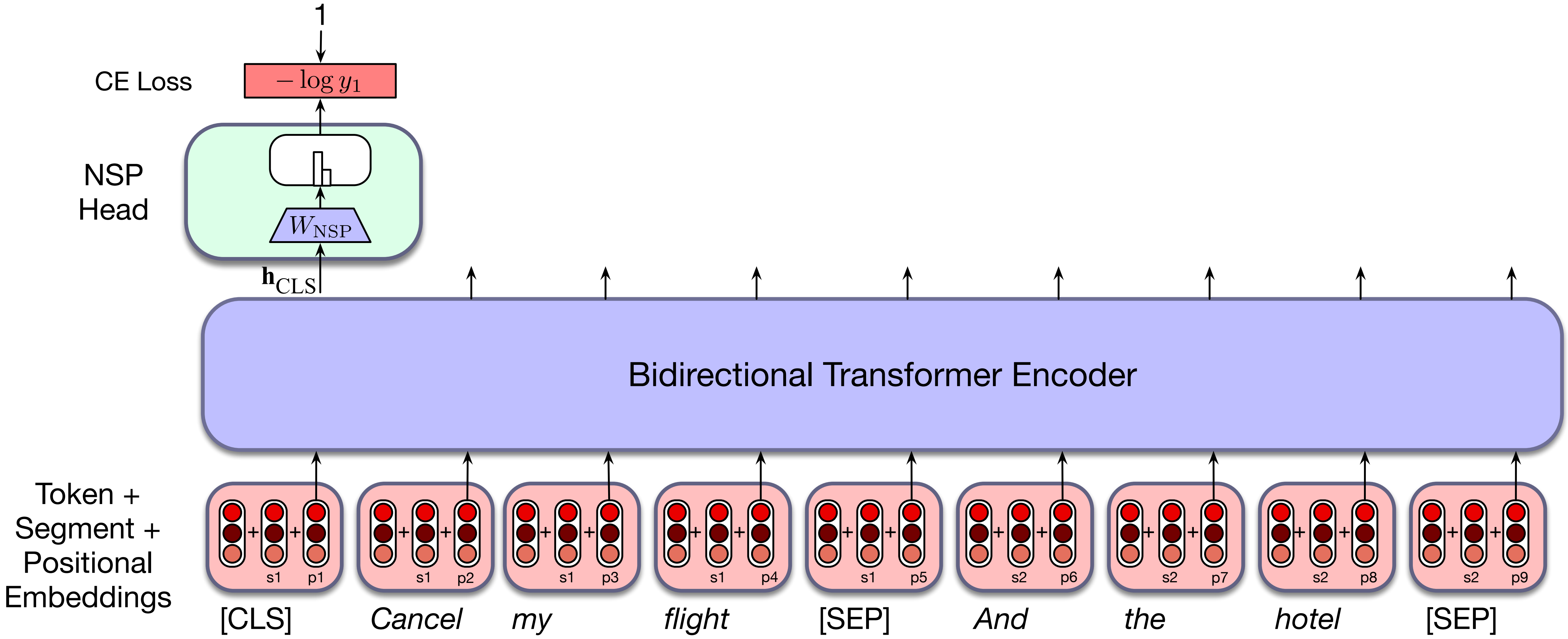
And two more special tokens

- [1st segment] and [2nd segment]
- These are added to the input embedding and positional embedding

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

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

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"

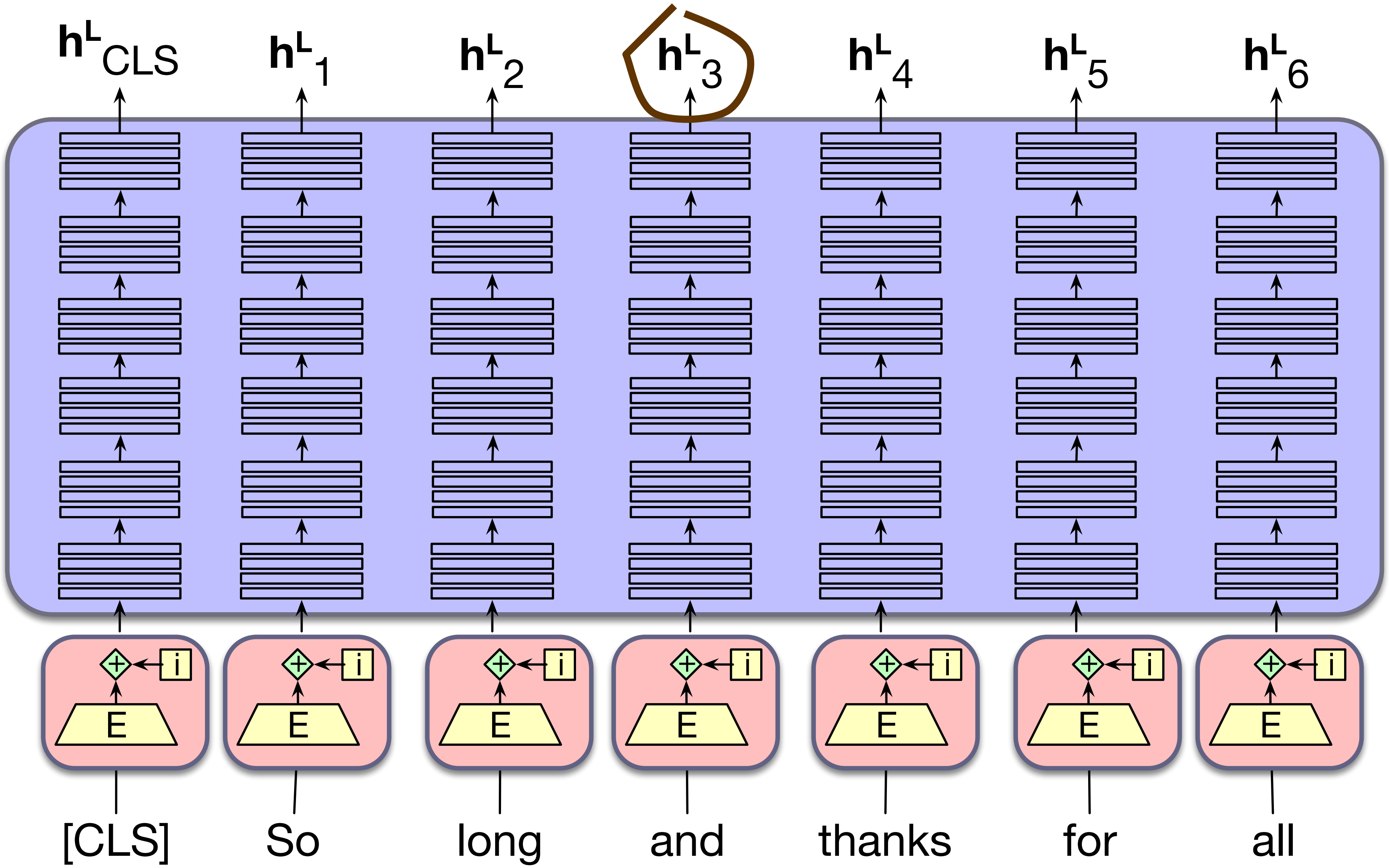
Masked Language Models

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)

Word sense

Words are ambiguous

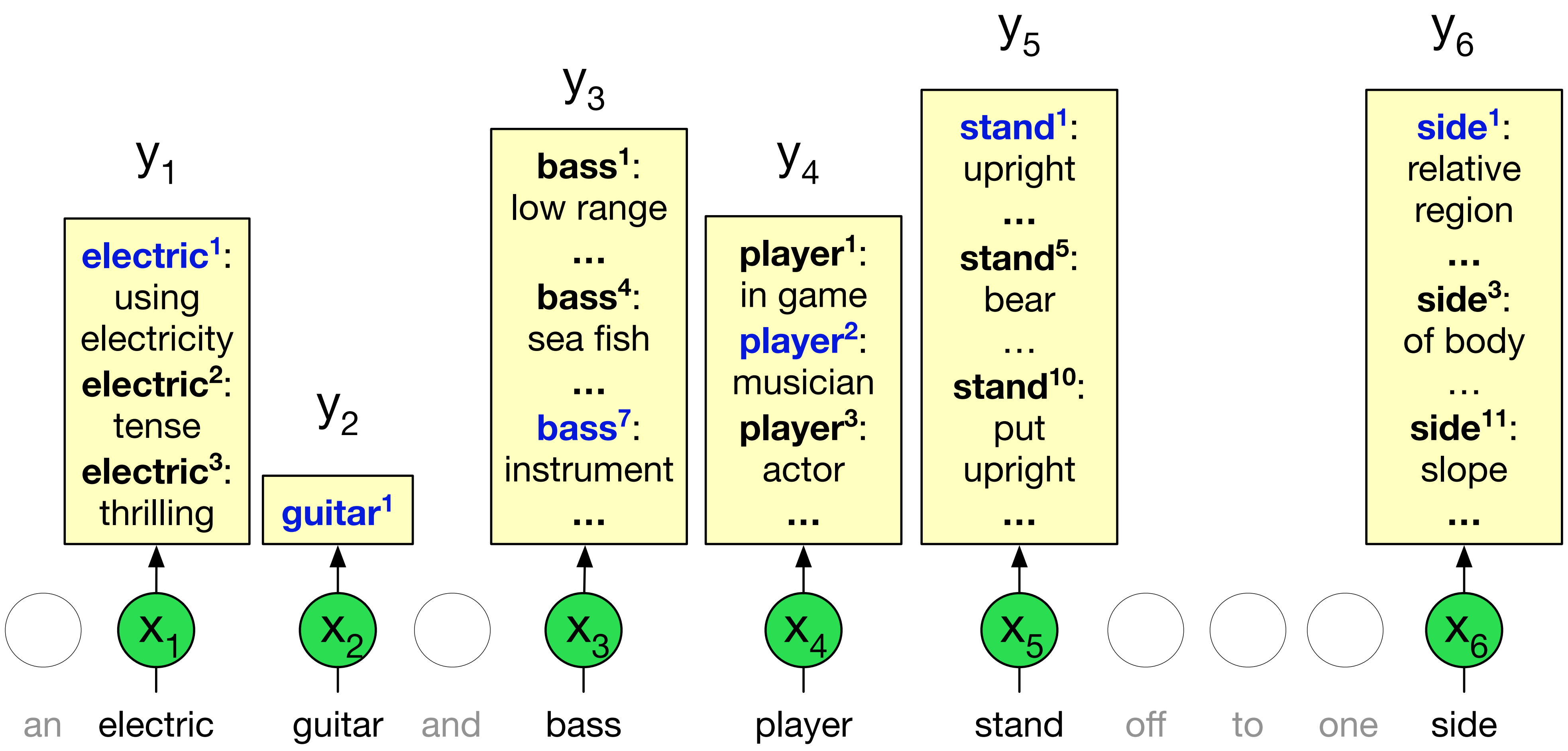
A **word sense** is a discrete representation of one aspect of meaning

mouse^1 : .... a *mouse* controlling a computer system in 1968.
mouse^2 : .... a quiet animal like a *mouse*
bank^1 : ...a *bank* can hold the investments in a custodial account ...
bank^2 : ...as agriculture burgeons on the east *bank*, the river ...

Contextual embeddings offer a continuous high-dimensional model of meaning that is more fine grained than discrete senses.

Word sense disambiguation (WSD)

The task of selecting the correct sense for a word.



1-nearest neighbor algorithm for WSD

Melamud et al (2016), Peters et al (2018)

At training time, take a sense-labeled corpus like SEMCOR

Run corpus through BERT to get contextual embedding for each token

- E.g., pooling representations from last 4 BERT transformer layer

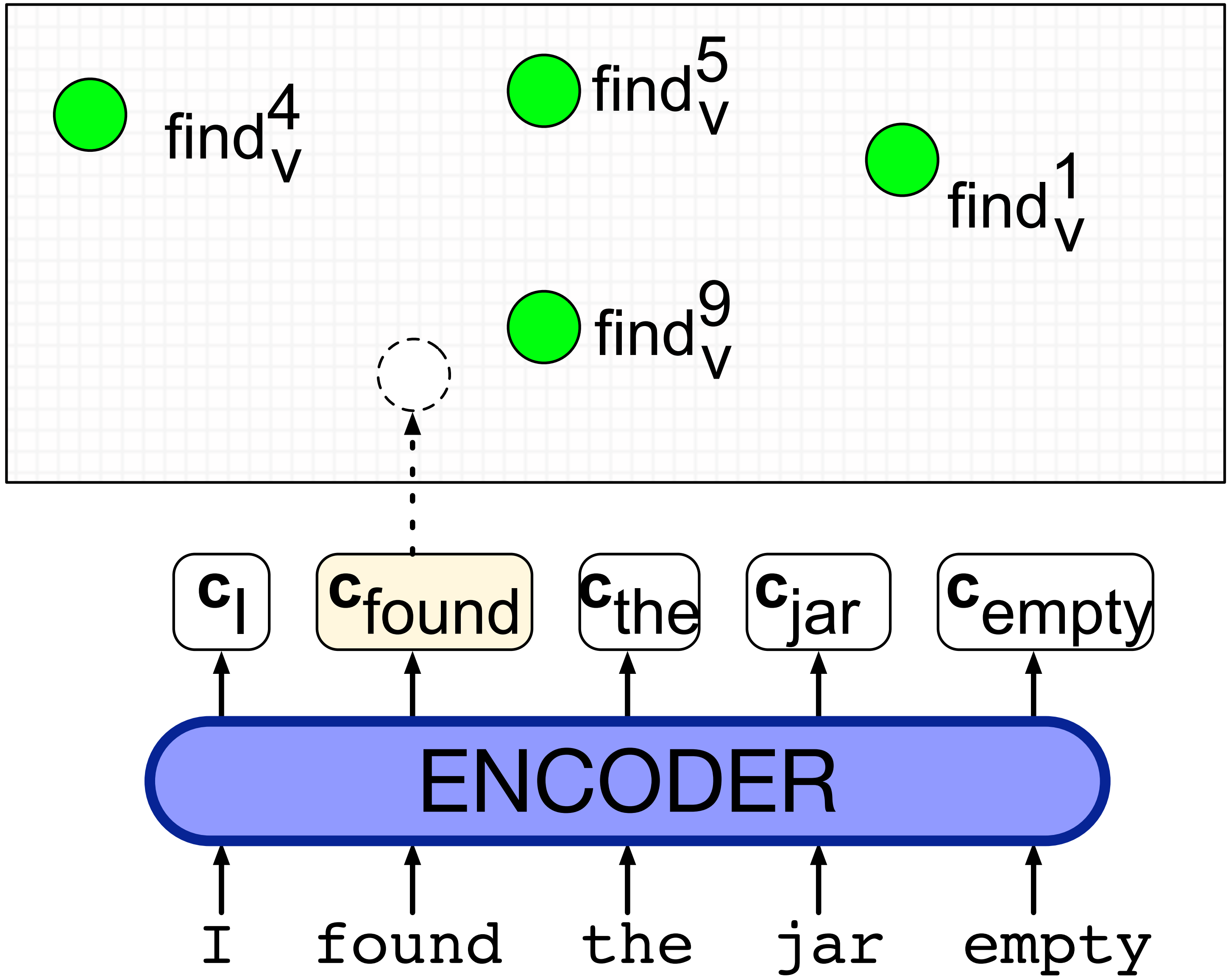
Then for each sense s of word w for n tokens of that sense, pool embeddings:

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

At test time, given a token of a target word *t*, compute contextual embedding t and choose its nearest neighbor sense from training set

$$\text{sense}(t) = \underset{s \in \text{senses}(t)}{\text{argmax}} \ \text{cosine}(\mathbf{t}, \mathbf{v}_s)$$

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)

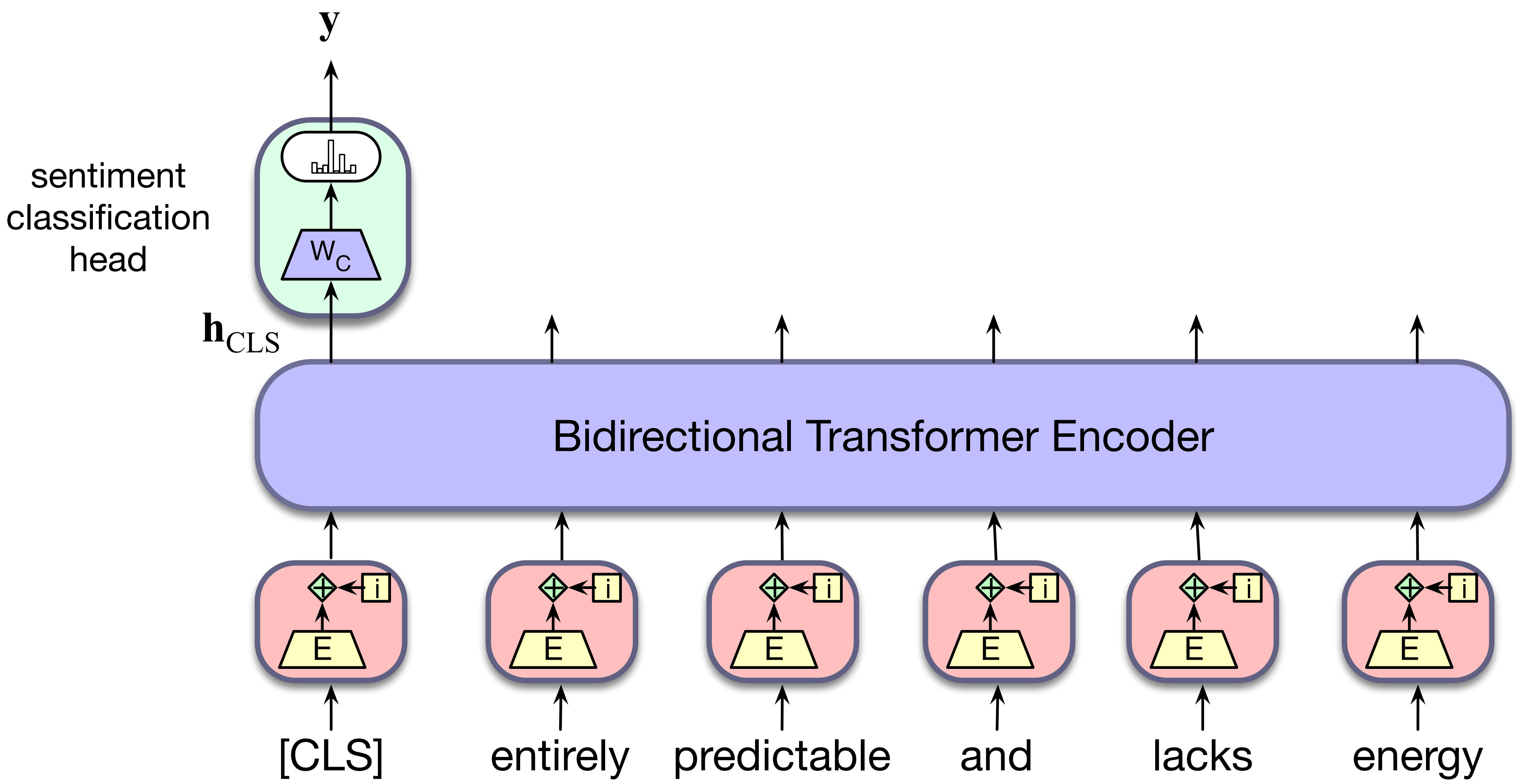
Masked Language Models

Contextual Embeddings

Masked Language Models

Fine-Tuning for Classification

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?)

Example: Natural Language Inference

Pairs of sentences are given one of 3 labels

- `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.

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 .

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

A **named entity** is anything that can be referred to with a proper name: a person, a location, an organization

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

Type	**Tag**	**Sample Categories**	**Example sentences**
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	**Mt. Sanitas** is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states	**Palo Alto** is raising the fees for parking.

Named Entity Recognition

Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

BIO Tagging

Ramshaw and Marcus (1995)

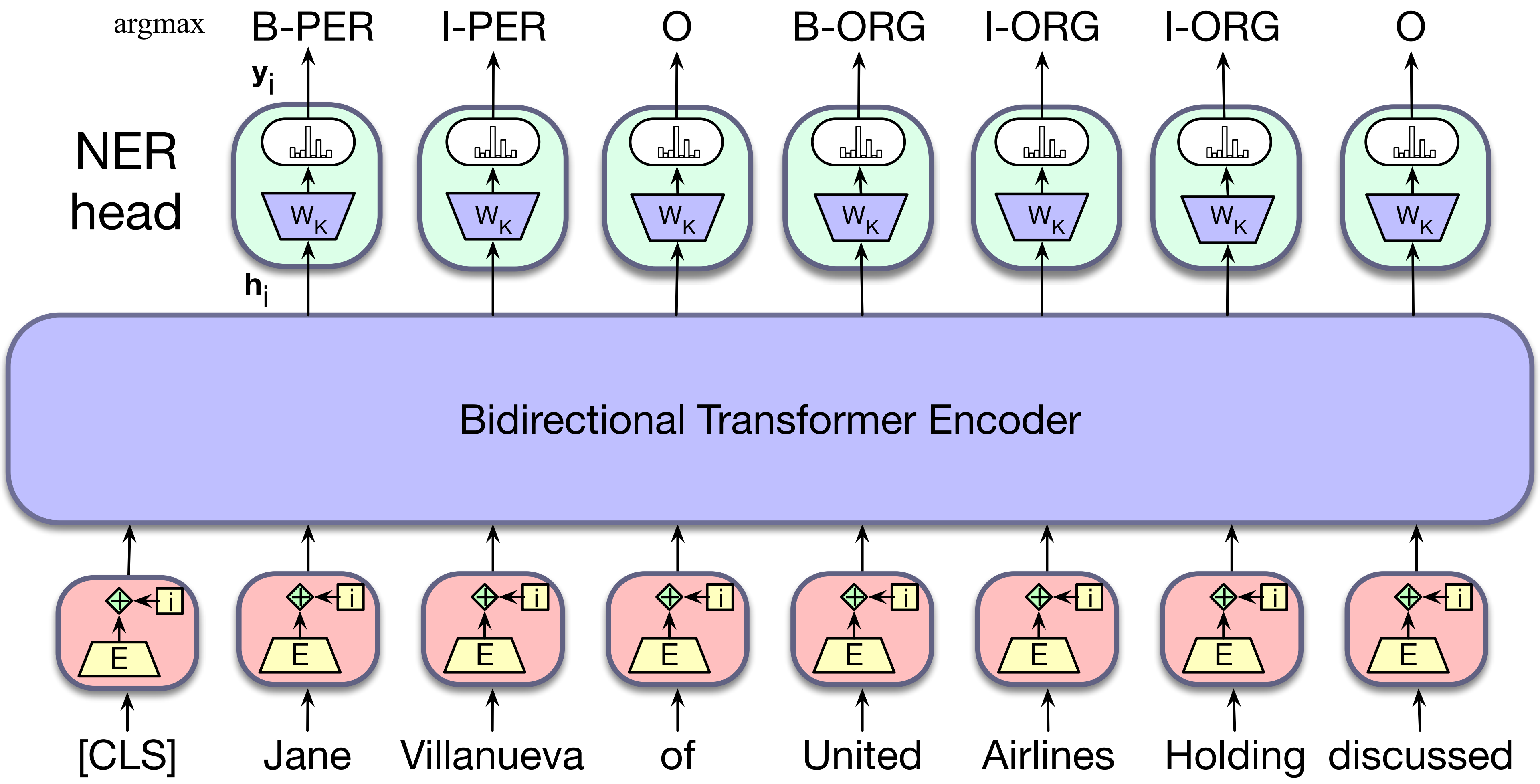
A method that lets us turn a segmentation task (finding boundaries of entities) into a classification task

[$_{PER}$ **Jane Villanueva**] of [$_{ORG}$ **United**] , a unit of [$_{ORG}$ **United Airlines Holding**] , said the fare applies to the [$_{LOC}$ **Chicago**] route.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Sequence labeling

$$\mathbf{y_i} = \text{softmax}(\mathbf{h_i^L W_K})$$
$$\mathbf{t_i} = \text{argmax}_k(\mathbf{y}_i)$$



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 Classification