Language Models are Unsupervised Multitask Learners

Presented by: Adam Zelzer, Nitzan Ron

OpenAI

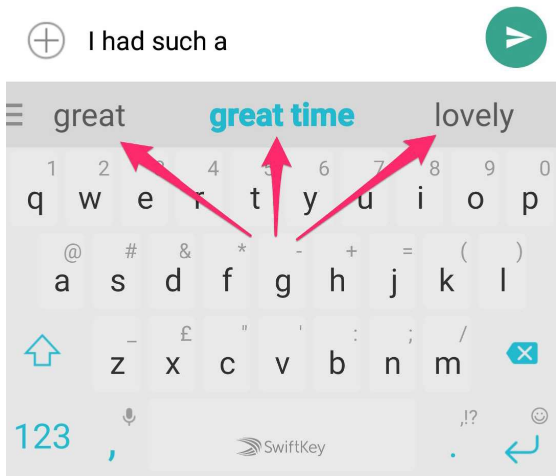
Introduction

About the Paper

- A paper by **OpenAI** employees and founders: Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever.
- Published in 2019.
- Describes GPT-II, the second generation of GPT models.

What is a Language Model?

A language model is a machine learning model that is able to predict the next word in a sentence based on its previous content.



Prior Work

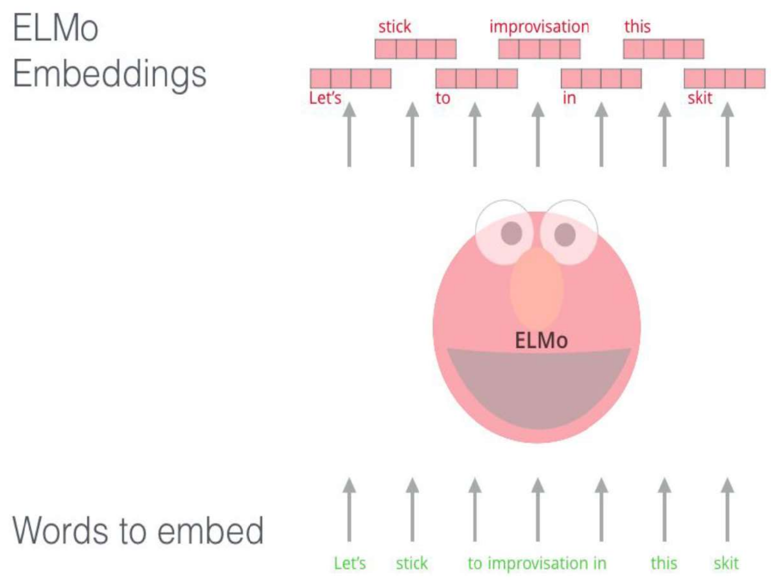
Previous Models

The field of NLP was relatively new at the time. There were a couple of different attempts and approaches to language modeling:

- ELMo (University of Washington, 2018)
- BERT (Google, 2018)
- MQAN (Salesforce, 2018)
- GPT-I (OpenAI, 2018)

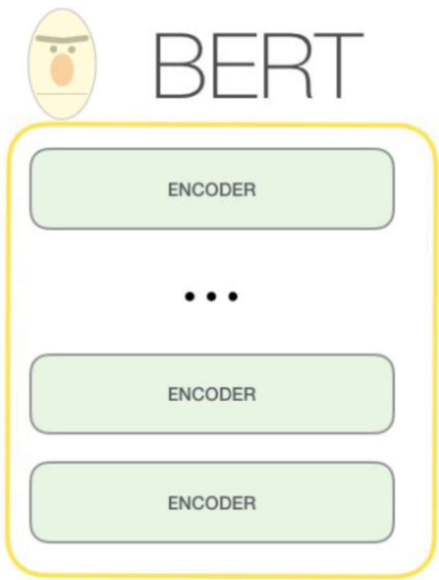
ELMo

- **E**mbeddings from **L**anguage **Mo**del.
- Based on an LSTM architecture.
- Provides context sensitive word embeddings.
- Fine-tuned for specific tasks.



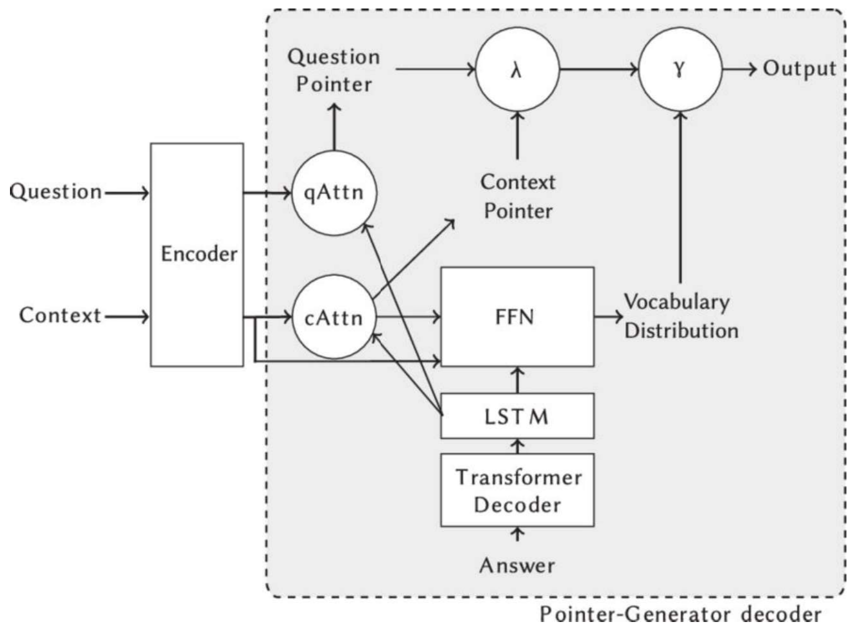
BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformer.
- Trained in a semi-supervised setting.
- Can be Fine-tuned for specific tasks, such as QA.



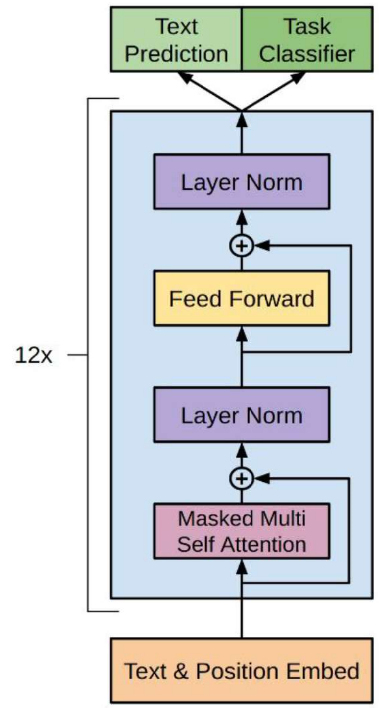
MQAN

- **M**ulti-task **Q**uestion **A**nswering **N**etwork.
- Utilizes both LSTMs and transformers.
- Multitask learner.
- Typically fine-tuned for specific tasks (QA).

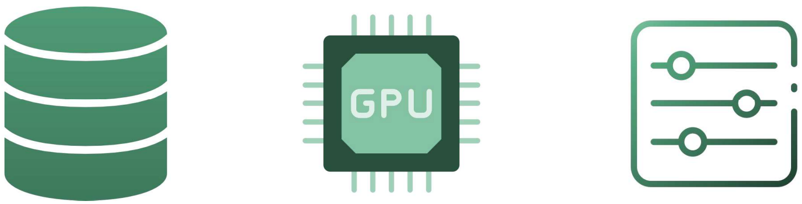


GPT-I

- **G**enerative **P**re-trained **T**ransformer.
- Predecessor to GPT-II.
- Trained in an unsupervised setting.
- General purpose model, able to generate coherent text.



Main Idea - MORE!



Language Modeling

Language is structured in a sequential ordering.

Given a sequence of symbols $x = (s_1, \dots, s_n)$, the model estimates the distribution:

$$p(x) = \prod_{i=1}^{n} p(s_i | s_1, \dots, s_{i-1})$$

Multitask Learning

Instead of learning: $p(output \mid input)$
We can learn: $p(output \mid input, task)$

This can be done easily with language modeling, using training examples of the form:

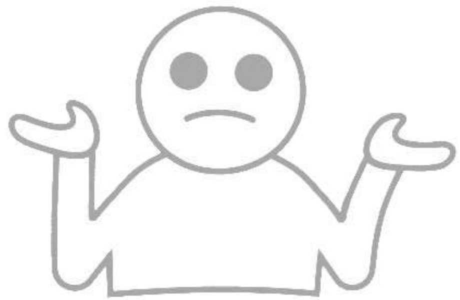
$$(task, input, output)$$

For example:

$$(\underbrace{\text{translate from language A to B}}_{\text{task}}, \underbrace{\text{<text in language A>}}_{\text{input}}, \underbrace{\text{<text in language B>}}_{\text{label}})$$

Required Dataset

To achieve multitask learning, a dataset whose objects contain task descriptions, text inputs and labels would be needed. There is no such dataset, and creating one is an immense job.



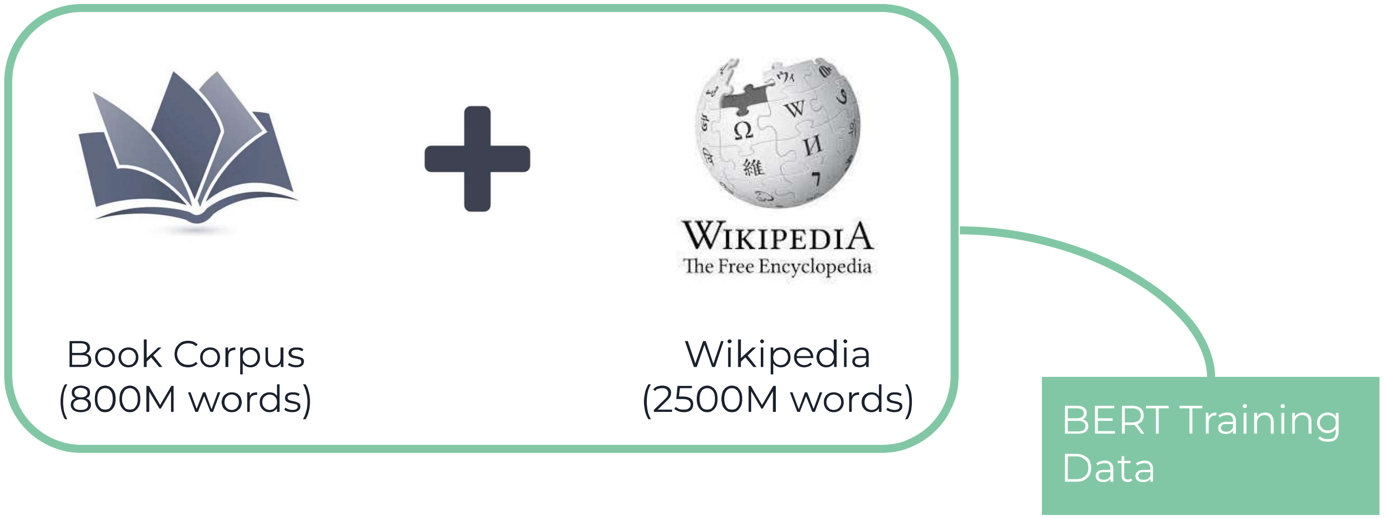
Solution - Unsupervised Learning

The researchers conjectured that an unlabeled, yet **large**, **high quality**, and **diverse** dataset would include enough language demonstration tasks in different domains that would help the model learn them.

This conjecture resulted in the creation of the **WebText** database.

Training Data - Prior Work

Most prior work trained language models on a single domain of text.



Common Crawl

Dataset containing web-page text, scraped from billions of pages.

- ✅ Large
- ✅ Diverse
- ❌ High-quality

WebText

Dataset containing Reddit pages whose **Karma** rating is 3 and above.

- Large
- Diverse
- High-quality

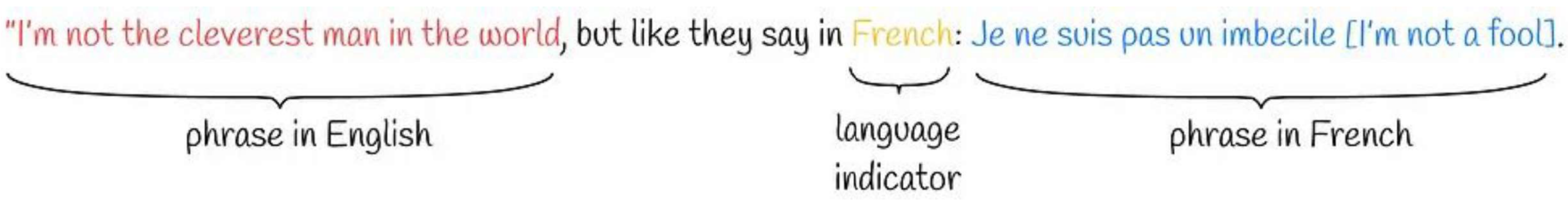
Resulting Dataset

Resulting dataset statistics (after deduplication):

- **45M** links
- **8M** documents
- **40GB** of text

Naturally Occurring Tasks

Example of naturally occurring demonstrations of different tasks found throughout the WebText training set:



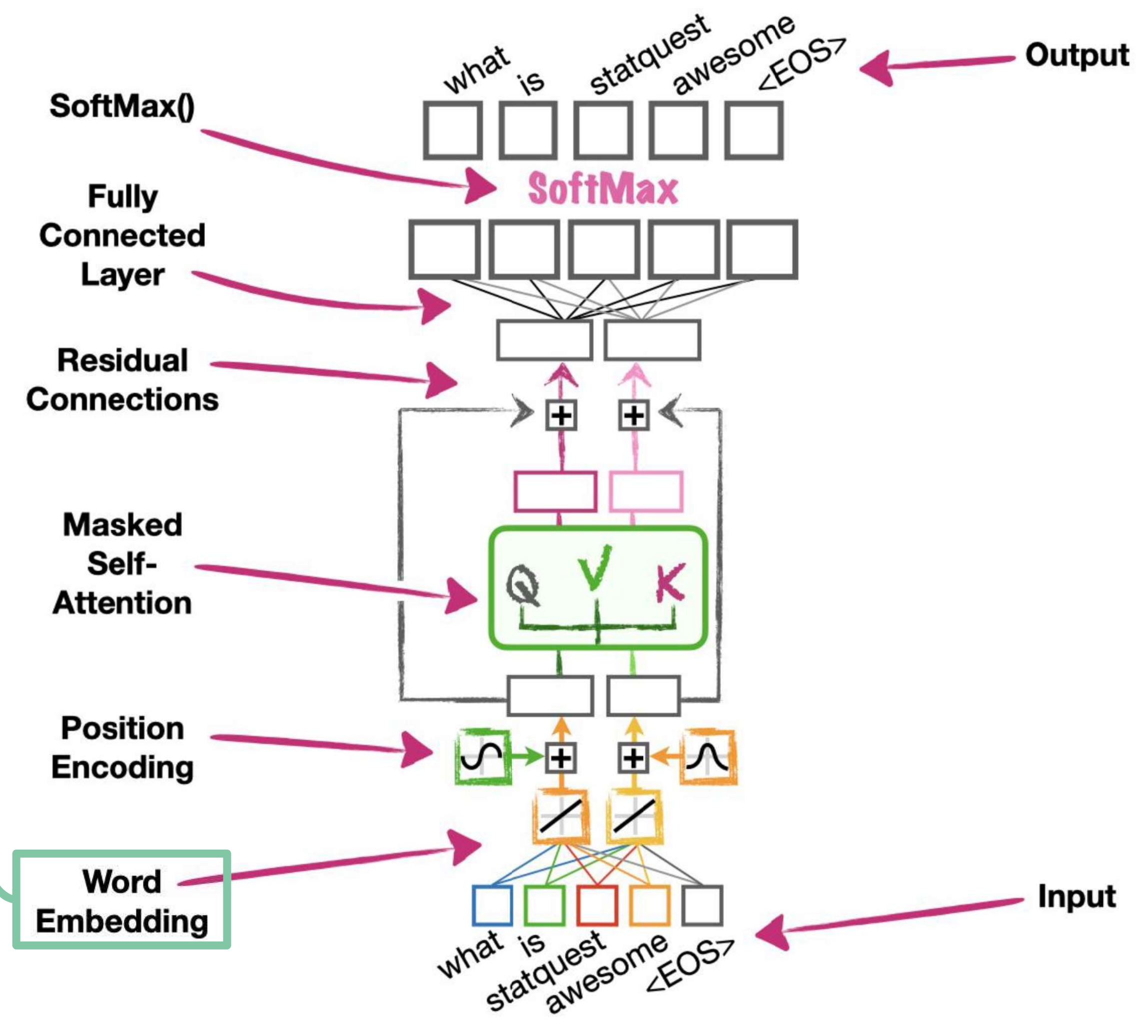
Model and details

Model

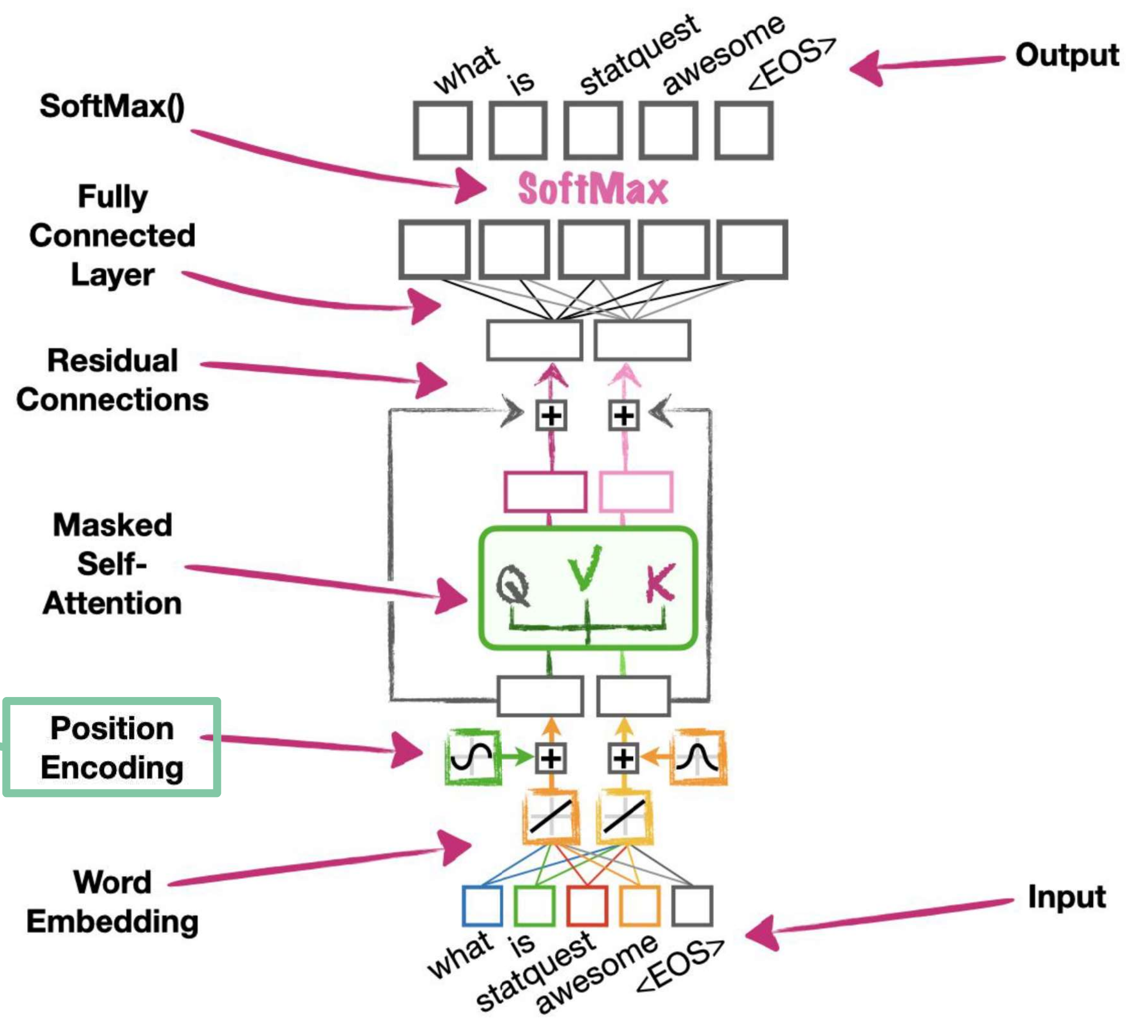
Like GPT-I, GPT-II uses a **Transformer** based model, which consists of the following:

- Word embedding
- Positional encoding
- Masked attention
- Feed-forward network
- Softmax

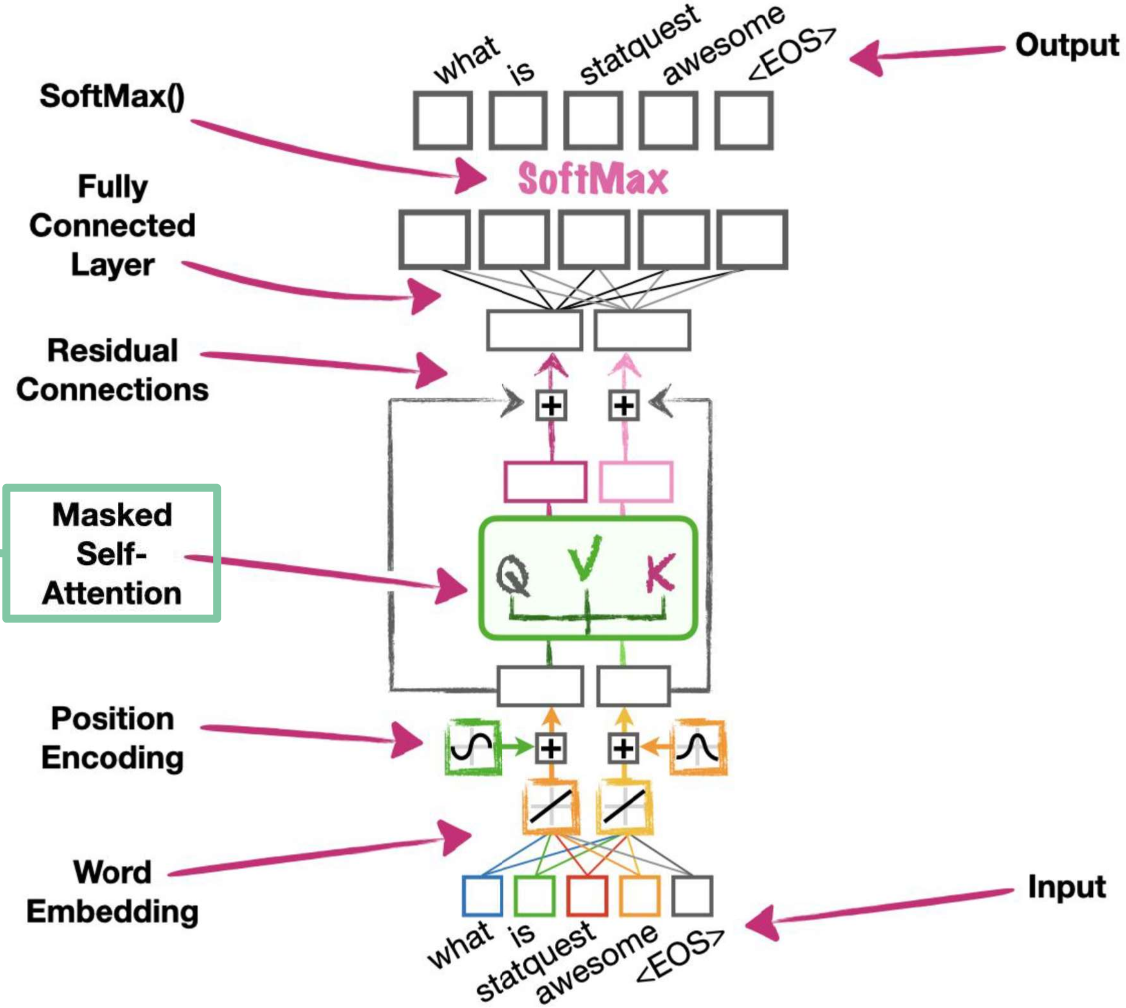
Embed words as vectors in a high-dimensional space, in a way that preserves meaning.



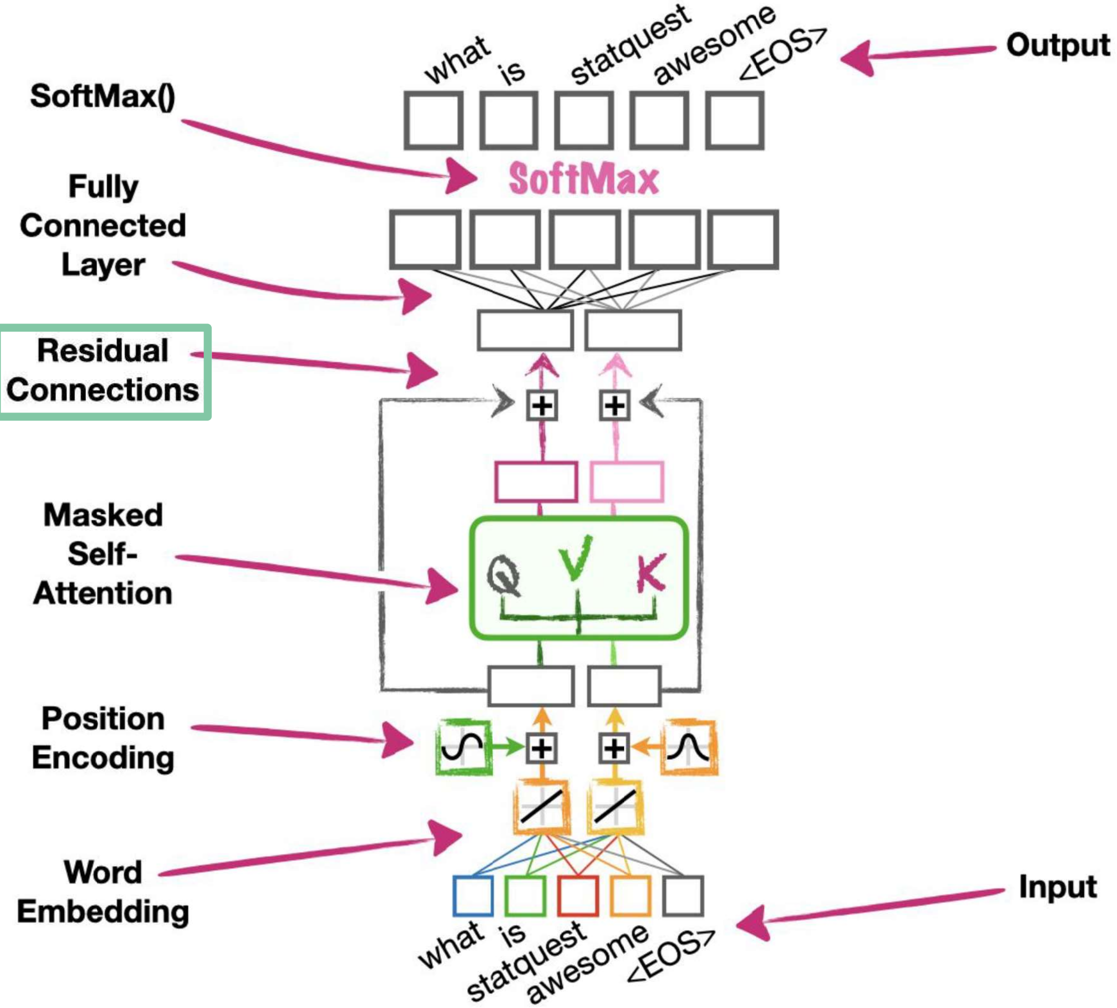
Encode position in sentence by adding sine waves at different frequencies and phases.



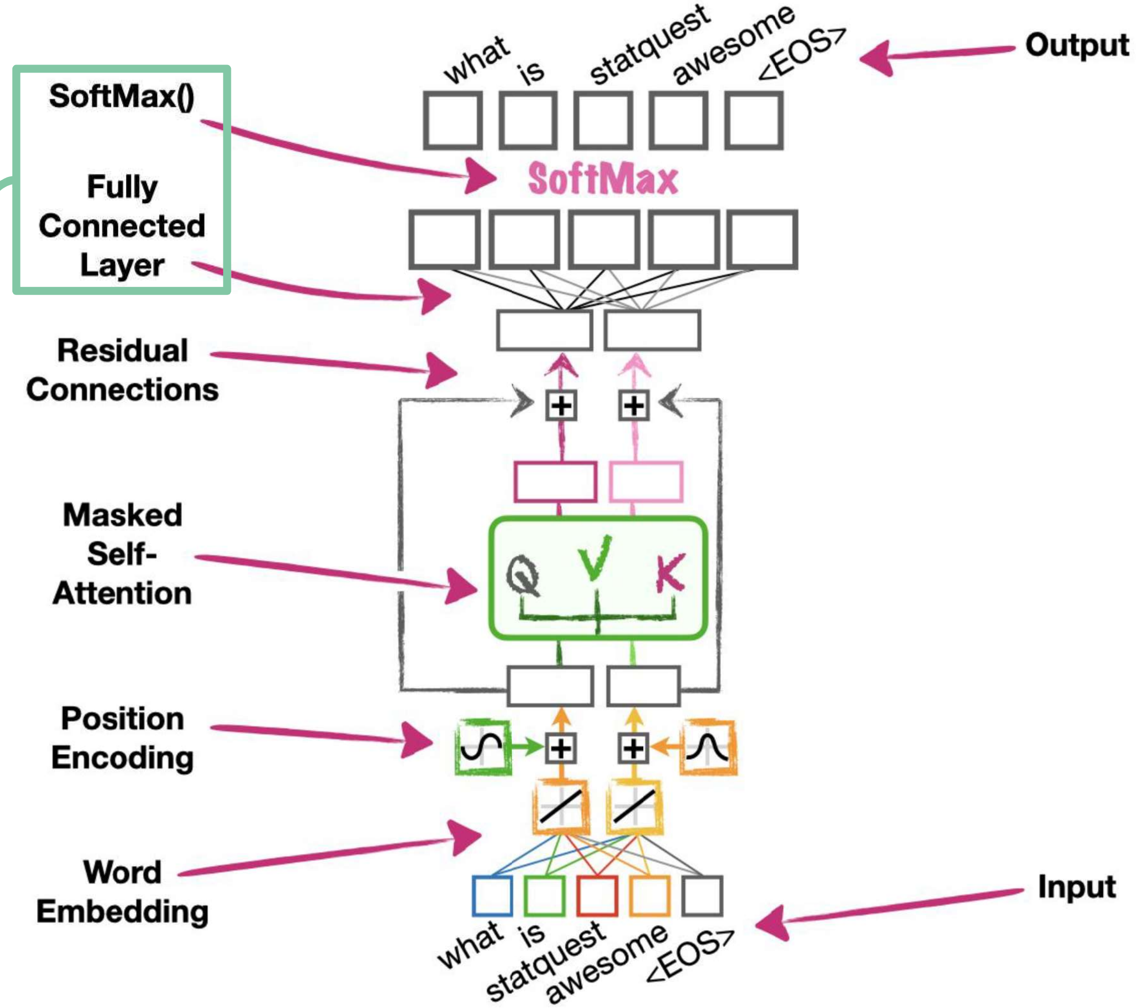
Apply self-attention to word vectors, masking future words, to attain contextual meaning.



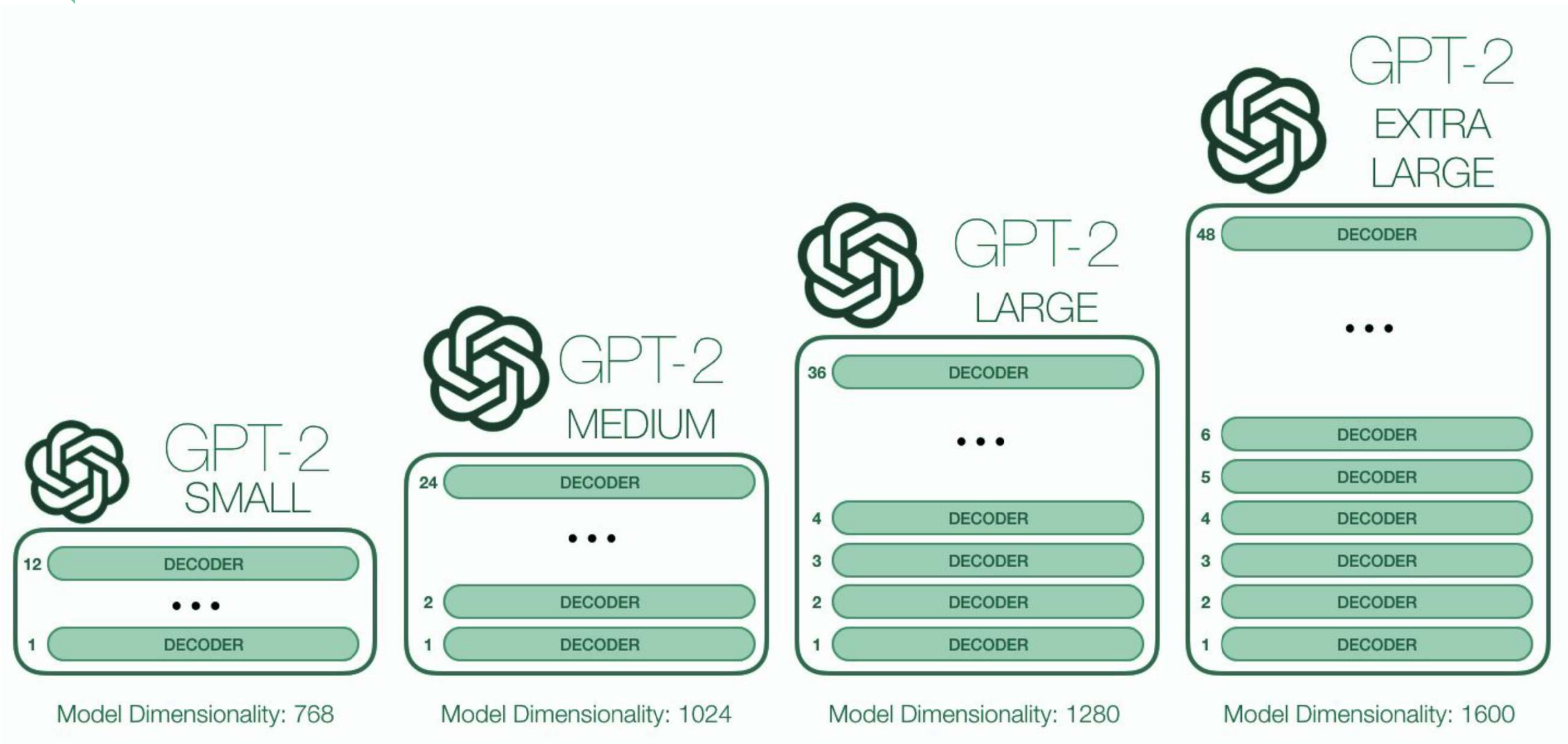
Add the attention vectors to positionally encoded vectors, to achieve the final word representation.



Pass through a fully connected network, to achieve a final predicted probability distribution.



GPT-II - Model Sizes



Compared to GPT-I

GPT-I is equivalent to the smallest size of GPT-II in terms of architecture, with the exception of the following:

- Layer normalization was moved to the input of each Transformer block.
- Context size was increased from 512 to 768.
- Batch size was augmented from 64 to 512.
- Vocabulary size was expanded from 40,000 tokens to 50,257.

Results

Perplexity

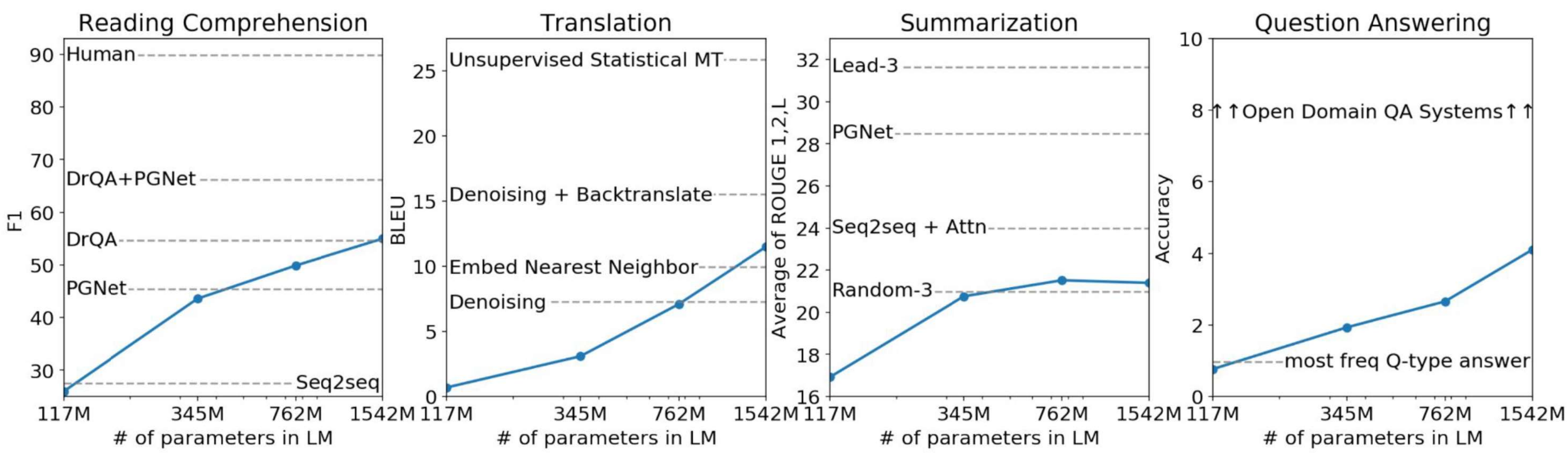
Perplexity measures a language model's uncertainty in predicting the next word, with lower values indicating better performance, the model predicts the data well.

$$Perp(x) = \frac{1}{\sqrt[n]{p(x)}}$$

Zero-Shot Results

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	WikiTex-t2 (PPL)	PTB (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	39.14	46.54	**21.8**
117M	**35.13**	45.99	**87.65**	**29.41**	65.85	75.20
345M	**15.60**	55.48	**92.35**	**22.76**	47.33	55.72
762M	**10.87**	**60.12**	**93.45**	**19.93**	**40.31**	44.575
1542M	**8.63**	**63.24**	**93.30**	**18.34**	**35.76**	42.16

Zero-Shot Results

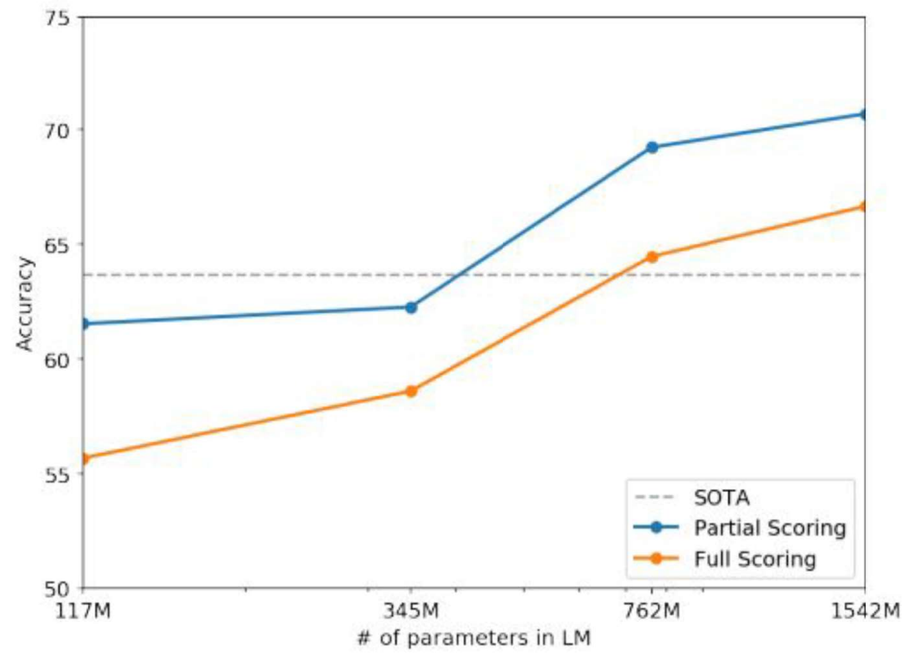


Winograd Schema Challenge

*“The **trophy** doesn’t fit into the brown suitcase because **it** is too large.”*

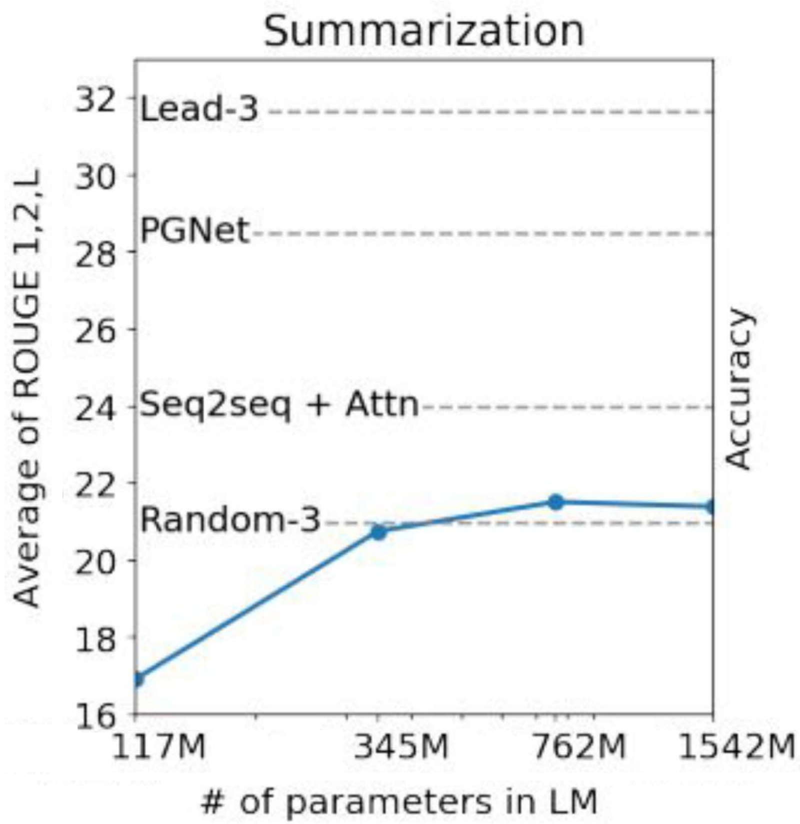
*“The trophy doesn’t fit into the brown **suitcase** because **it** is too small.”*

In both cases, the model succeeded.



Summarization

- Added text **TL; DR:** after the article and generated 100 tokens with Top 2 random sampling.
- Utilized **CNN** and **Daily Mail** datasets.
- Used 3 generated sentences from these 100 tokens to create the summary.

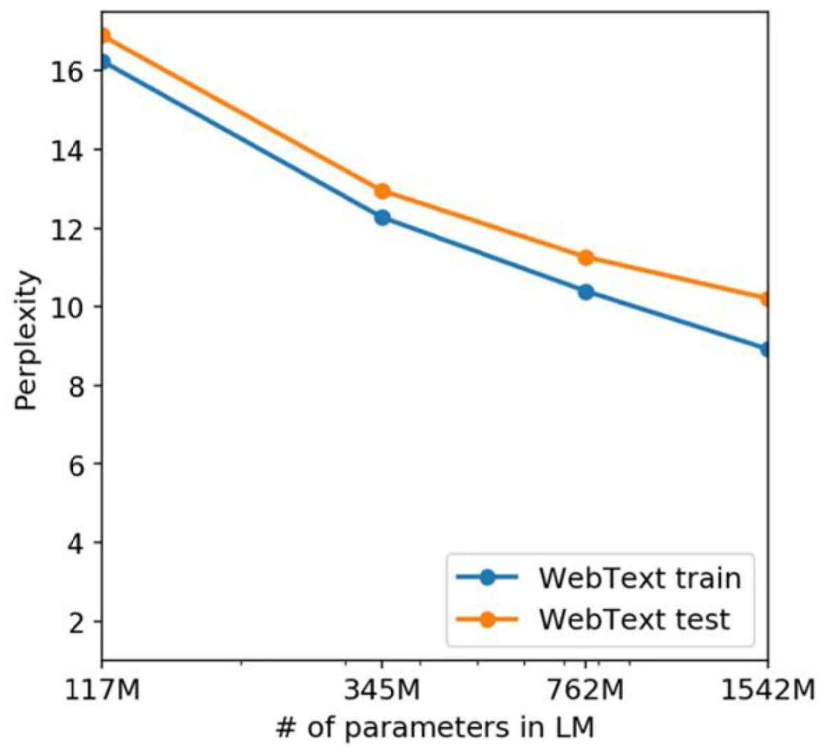


Generalization vs. Memorization

It is important to analyze how much test data also shows up in the training data to assess the generalization error accurately.

	PTB	WikiText-2	enwik8	text8	WikiText-103	1BW
Dataset train	**2.67%**	0.66%	**7.50%**	2.34%	**9.09%**	**13.19%**
WebText train	0.88%	**1.63%**	6.31%	**3.94%**	2.42%	3.75%

WebText Underfitting



What’s next?

Practical Performance

While the model is qualitatively performing the tasks, its performance is still only rudimentary according to quantitative metrics. In terms of practical applications, the zero-shot performance of GPT-II is still far from usable, and often no better than random for many tasks.

In addition, many other practical tasks remain to be evaluated.

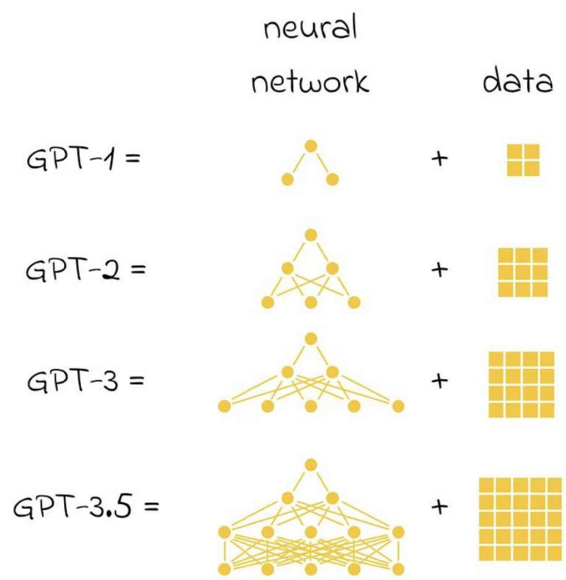
Fine-tuning

The work done so far did not include fine-tuning for specific tasks and purposes. Thus, the potential with fine-tuning remains unclear.

The researchers plan to explore fine-tuning on different benchmarks in order to improve results.

Our thoughts

More parameters seem to improve the model substantially, so it is natural to try increasing model size.



Conclusions

Summary and Conclusions

- A LLM trained on a sufficiently large and diverse dataset is able to perform well across many domains.
- GPT-II demonstrates SOTA performance on 7 out of 8 tested language modeling datasets.
- Maximizing the likelihood of a sufficiently varied text corpus allows a model to learn how to perform many tasks without the need for explicit supervision.

A thought-provoking question

OpenAI didn’t release the code for the GPT-II model.

Why do you think that is?

Do you agree with their choice?