Claim 1 of US11556786B2 vs. disclosure in language_understanding_paper.pdf

#	Claim Limitation	Supporting Text	Explanation	Support
1.0	**A method of generating an output sequence comprising a plurality of output tokens from an input sequence comprising a plurality of input tokens the method comprising at each of a plurality of generation time steps**	Page 2: This model applies a multi-headed self-attention operation over the **input context tokens** followed by position-wise feedforward layers to **produce an output distribution over target tokens**. Page 6: For RACE (question answering), we pick the answer the generative model assigns the highest average token log-probability when conditioned on the **document and question**. Page 6: For SST-2 (sentiment analysis), we append the token very to each example and restrict the language model's output distribution to only the words positive and negative and guess the token it assigns higher probability to as the prediction.	The reference discloses a generative model that produces output distributions over tokens and uses those distributions to select specific output tokens for downstream tasks. While the excerpts describe applying the trained model to specific tasks rather than pure sequence generation, they demonstrate the model's ability to generate probability distributions over tokens from input contexts and select output tokens based on those distributions. The 'output distribution over target tokens' combined with the task-specific examples of token selection shows the model's fundamental capability to generate and select output tokens from input sequences, even if the primary focus is on task adaptation rather than open-ended generation.	Partial
1.1	**generating a combined sequence for the generation time step that includes the input sequence followed by the output tokens that have already been generated as of the generation time step**	Page 2, Section 3.1: 'Given an unsupervised corpus of tokens U={u_1,...,u_n}, we use a standard language modeling objective to maximize the following likelihood: **L_1(U)=sum_i log P(u_i | u_{i-k},...,u_{i-1}; Theta)**' Page 2, Section 3.1: 'This model applies a multi-headed self-attention operation over **the input context tokens** followed by position-wise feedforward layers to produce **an output distribution over target tokens**'	The reference describes a language model that conditions on preceding tokens (u_{i-k}...u_{i-1}) to predict the next token, which inherently involves processing a sequence that includes previously generated tokens. However, the reference does not explicitly disclose an iterative generation process where output tokens are concatenated with the original input sequence to form a combined sequence for each generation step. The language modeling objective suggests the model processes context that could include prior outputs, but this is presented as standard language modeling rather than the specific concatenation mechanism claimed.	Inferential
1.2	**processing the combined sequence using a self-**	Excerpt 1: Page 2, Section 3.1: 'Given an unsupervised corpus of tokens U={u_1,...,u_n}, we use a standard language modeling objective to	The reference discloses a decoder-only transformer that uses masked self-attention to process input context tokens and generate output distributions. While the language	Inferential

#	Claim Limitation	Supporting Text	Explanation	Support
	attention decoder neural network	maximize the following likelihood: L_1(U)=sum_i log P(u_i | u_{i-k},...,u_{i-1}; Theta)' Excerpt 2: Page 2, Section 3.1: 'This model applies a **multi-headed self-attention operation over the input context tokens** followed by position-wise feedforward layers to produce an output distribution over target tokens' Excerpt 3: Page 4: 'We trained a **12-layer decoder-only transformer with masked self-attention heads** (768 dimensional states and 12 attention heads)' Excerpt 4: Page 2, Section 3.1: 'In our experiments, we use a **multi-layer Transformer decoder** [34] for the language model, which is a variant of the transformer [62]'	modeling objective shows the model conditions on preceding tokens (u_{i-k}...u_{i-1}) to predict the next token, the excerpts do not explicitly describe concatenating input sequences with previously generated output tokens at each generation step. The 'input context tokens' could refer to the original input sequence rather than a dynamically combined sequence. The decoder-only architecture with masked self-attention is consistent with processing sequences that include prior outputs, but this specific concatenation mechanism is not explicitly disclosed in the cited excerpts.	
1.3	**wherein the self-attention decoder neural network comprises a plurality of neural network layers that include a plurality of masked self-attention neural network layers**	We trained a **12-layer decoder-only transformer with masked self-attention heads** (768 dimensional states and 12 attention heads). This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feedforward layers to produce an output distribution over target tokens. In our experiments, we use a **multi-layer Transformer decoder** [34] for the language model, which is a variant of the transformer [62].	The reference explicitly discloses a 12-layer decoder-only transformer architecture where masked self-attention heads are distributed across multiple layers. In transformer architectures, each layer contains multiple attention heads, and the specification of 'masked self-attention heads' within a '12-layer decoder-only transformer' indicates that masked self-attention mechanisms are present across multiple neural network layers. The 'multi-layer Transformer decoder' description combined with 'masked self-attention heads' confirms that the plurality of neural network layers (12 layers) includes masked self-attention components distributed throughout the layer stack, satisfying the claimed limitation of 'a plurality of neural network layers that include a plurality of masked self-attention neural network layers.'	Explicit
1.4	**and wherein the self-attention decoder neural network is configured to process**	Page 2, Section 3.1: 'where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Theta'	The reference discloses a multi-layer Transformer decoder that processes sequences through 12 neural network layers (h_l = transformer_block(h_{l-1})) to generate	Partial

#	Claim Limitation	Supporting Text	Explanation	Support
	the combined sequence through the plurality of neural network layers to generate a time step output that defines a score distribution over a set of possible output tokens;	Page 2, Section 3.1: '**h_l = transformer_block(h_{l-1}) for all i in [1,n]**' Page 2, Section 3.1: '**P(u) = softmax(h_n W_e^T)**' Page 2: 'This model applies a **multi-headed self-attention operation** over the **input context tokens** followed by position-wise feedforward layers **to produce an output distribution over target tokens**' Page 6: 'we pick the answer the generative model assigns the highest average token log-probability'	probability distributions (P(u) = softmax(h_n W_e^T)) over target tokens. The language modeling objective conditions on preceding context tokens (u_{i-k}...u_{i-1}), which inherently represents a combined sequence of input plus previously generated tokens. The 'output distribution over target tokens' produced through multi-layer processing functions as a score distribution, as demonstrated by task-specific examples where the model assigns probabilities to tokens for selection based on these distributions.	
1.5	**and selecting using the time step output an output token from the set of possible output tokens as the next output token in the output sequence.**	Page 2, Section 3.1: 'Given an unsupervised corpus of tokens U={u_1,...,u_n}, we use a standard language modeling objective to **maximize the following likelihood: L_1(U)=sum_i log P(u_i | u_{i-k},...,u_{i-1}; Theta)**' Page 2, Section 3.1: '**P(u)** = **softmax(h_n W_e^T)**' Page 6: 'For DPRD [46] (winograd schemas), we replace the definite pronoun with the two possible referrents and **predict the resolution that the generative model assigns higher average token log-probability to the rest of the sequence after the substitution.**'	The reference discloses the fundamental language modeling process where tokens are predicted sequentially based on preceding context (u_{i-k}...u_{i-1}), with each token selection determined by probability distributions generated through softmax. The language modeling objective inherently involves iterative token selection where each selected token becomes part of the context for predicting subsequent tokens. While the DPRD example shows task-specific application, it demonstrates the model's ability to evaluate token probabilities and continue sequences based on those selections. The combination of the language modeling objective with the softmax probability generation establishes the core mechanism of selecting output tokens from probability distributions for sequence generation.	