CBAG: Conditional Biomedical Abstract Generation

Justin Sybrandt [*] [1] and Ilya Safro [†] [1]

[1] School of Computing, Clemson University

Abstract

Biomedical research papers use significantly different language and jargon when compared to typical English text, which reduces the utility of pre-trained NLP models in this domain. Meanwhile Medline, a database of biomedical abstracts, introduces nearly a million new documents per-year. Applications that could benefit from understanding this wealth of publicly available information, such as scientific writing assistants, chat-bots, or descriptive hypothesis generation systems, require new domain-centered approaches. A conditional language model, one that learns the probability of words given some a priori criteria, is a fundamental building block in many such applications. We propose a transformer-based conditional language model with a shallow encoder "condition" stack, and a deep "language model" stack of multi-headed attention blocks. The condition stack encodes metadata used to alter the output probability distribution of the language model stack. We sample this distribution in order to generate biomedical abstracts given only a proposed title, an intended publication year, and a set of keywords. Using typical natural language generation metrics, we demonstrate that this proposed approach is more capable of producing non-trivial relevant entities within the abstract body than the 1.5B parameter GPT-2 language model. **Reproducability:** All code, data, pre-trained models, and experimental parameters are available online: sybrandt.com/2020/cbag

1 Introduction

The biomedical sciences are becoming more data driven due to the increased availability of experimental data and the democratization of machine learning algorithms. One subfield of biomedical data science, literature-based discovery [35], produces algorithms to automatically identify plausible research directions from the growing body of scientific literature [8]. While these systems have seen early successes aiding biomedical science [1, 4], these techniques often lack the interpretability necessary to persuade domain scientists to pursue algorithmically generated leads. While customizable visualizations aid significantly [32], many researchers would prefer a textual description to accompany generated hypotheses. Thinking much further into the future, if hypothesis generation systems are ever going to function as automated scientists in their own right, they will require the ability to generate textual arguments supporting their own ideas.

Today, modern deep-learning language generation models can produce text in a range of contexts. Building off of the transformer architecture [36], models like BERT [10] and GPT/GPT-2 [28, 29] have set a new standard in a range of natural language benchmarks [39]. Adaptations of these models, such as SciBert [5] and BioBert [20], have retrained the baseline models for domain-specific tasks in order to advance the state of domain-specific benchmarks as well [15, 26]. However, these domain-specific models are not designed for natural language generation (NLG). Models like GPT are trained to generated text in the language of typical English writing, and we demonstrate below that these generations are ill-suited for the jargon-filled particular language used by biomedical scientists.

Compounding these challenges around generating biomedical text, much less work has focused on *conditional* generation, wherein the output language distribution is affected by a priori knowledge. Older text captioning systems [41] follow a similar approach using sequence models informed by image encodings. However, more control over generated text is necessary for applications like hypothesis generation systems, where

[*] jsybran@clemson.edu
[†] isafro@clemson.edu

semantic information detected by the system should be leveraged in an automatically produced argument. Modern systems trained outside the biomedical domain, such as Ctrl [17] allow for some conditions, but lack the flexibility needed to capture sets of semantic information. More generalizable methods, such as those produced by variational auto-encoders [16], can capture rich latent language semantics, but cannot straightforwardly encode domain-based information, such as a set of keywords one wishes to include in the output text.

A language model that can enable complex domain-specific applications, such as hypothesis generation, therefore requires a new approach. This technique should accept an arbitrary set of semantic criteria as a condition, should be aware of domain-specific entities and jargon, and should produce text that would be expected by biomedical scientists.

In this paper we propose CBAG, a conditional biomedical abstract generation model that seeks to address the above requirements. This transformer model includes a shallow encoder stack to encode qualities of the condition, and an deep decoder stack to produce a high quality language model. We train this model using semi-supervised multi-task generative pre-training, wherein to minimize our proposed objective function, the model must predict successive tokens, parts of speech, dependency tags, as well as entity labels. We train this model using over 20-million biomedical records provided by the National Library of Medicine (NLM) through the Medline database. Each record consists of a title, abstract, publication year, and an optional set of author-provided keywords. Text processing and annotations are provided by a biomedical NLP model trained on the "BIONLP13CG" BioCreative training set [15]. This pre-trained domain-specific model allows the CBAG model to apply the knowledge gain from the relatively small human-annotated dataset to the larger set of unstructured text present in Medline.

We train the proposed model by sampling textual windows from within MEDLINE abstracts. The publication date, and any author-supplied Medical Subject Headings (MeSH terms, a set of biomedical keywords and phrases) form the condition. The sampled window serves as input to the decoder stack. Windows are split into subword units using the unigram subword-regularization algorithm [18]. Using masked-self attention, we train the model to predict each subword $i + 1$ using only the condition and tokens $1, \dots, i$.

To the best of our knowledge, this work is the first attempt to design a biomedical abstract generator. Therefore, without a direct point of comparison, we leverage the 1.5-billion parameter "huge" version of GPT-2 to compare against. As this language model was trained on a range of online data sources, such as the BooksCorpus and English Wikipedia, it is a disadvantage in our domain-specific task. However, the authors find that this model is capable of a range of specific tasks across domains, such as language translation, question answering, and commonsense reasoning [29]. Furthermore, other work has even found that the GPT-2 language model can function as a general purpose knowledge base [27]. For these reasons, we can expect GPT-2 to be a relevant, albeit disadvantaged, point of comparison.

When generating an abstract during evaluation, we formulate a human written title, as well as relevant condition information where applicable, for model input. We then sample each model's subword probability distribution for each generated result until the new abstract is written. We evaluate computer-generated abstracts based on their ability to produce relevant n-grams that occur in the human-written abstract associated with the input title. We leverage a range of NLG metrics [31], such as Bleu, METEOR, ROUGE-L and CIDEr, including a version of CIDEr that omits input n-grams from consideration. Through all considered metrics we quantitatively demonstrate increased performance through the use of CBAG. Qualitatively, we present full-abstracts, as well as a handful of sentences for assorted generations, which show the ability of our proposed model to capture the overarching flow of scientific summaries. We additionally demonstrate the ability for condition keywords to influence model generations by producing a varied set of completions for the seed-phrase, "*In this study, we found...*"

Our contribution: We present CBAG, a transformer-based language model for conditional biomedical abstract generation. Trained using Medline records and informed by semi-supervised domain-specific annotations, this model captures biomedical jargon, entities, and pattern of scientific discussion. We compare generated abstracts against the 1.5B parameter GPT-2 language model, and demonstrate a superior ability to produce relevant n-grams across a range of NLG metrics.

All code, data, pre-trained models, preprocessing pipelines, and experimental parameters are available online[1]. We additionally supply a set of over 13,000 automatically generated abstracts for a wide range

[1] sybrandt.com/2020/cbag

of test-set titles. Using the generalizable precondition approach presented here, we hope to enable future applications, such as descriptive hypothesis generation. However, we are also cognisant of the potential for abuse surrounding high quality domain-specific language models. We discuss these concerns further in Section 7.

2 Background

While recent **language models** receive a newfound popularity in proportion to their surprising capacity across a range of tasks [29], their study predates modern machine learning techniques [6]. Formally, a language model is a probabilistic model that captures the conditional probability of each next element in a sequence given all prior elements. Specifically, this is described by the function:

$$\mathsf{Pr}(s) = \prod_{i=1}^{n} \mathsf{Pr}(s_i | s_1, \dots, s_{i-1}) \tag{1}$$

Here, s is a sequence of n elements. The probability of observing sequence s is determined by the product of the conditional probabilities of observing each token s_i given all prior tokens. These models can generate new text by iteratively sampling new elements from the probability distribution $\mathsf{Pr}(s_{i+1}|s_1, \dots, s_i)$.

The conditional language model introduces a new term c into the above equation. The condition can allow applications to alter the resulting sequence based on a priori knowledge [16]. Formally, the conditional language model is defined as:

$$\mathsf{Pr}(s|c) = \prod_{i=1}^{n} \mathsf{Pr}(s_i | s_1, \dots, s_{i-1}, c) \tag{2}$$

Modern neural network language models [29, 17], model these probability distributions by minimizing the negative log-likelihood of these distributions over a large training set of sequences:

$$\begin{aligned} &\mathcal{L}\left(\left(s^{(1)}, c^{(1)}\right), \dots, \left(s^{(m)}, c^{(m)}\right)\right) \\ &= -\sum_{j=1}^{m}\sum_{i=1}^{n} \log \mathsf{Pr}_\theta \left(s_i^{(j)} | s_1^{(j)}, \dots, s_{i-1}^{(j)}, c^{(j)}\right) \end{aligned} \tag{3}$$

Here, Pr_θ indicates the parameterized model that approximates the language model distribution. Modern systems often use the transformer architecture [29, 17, 38] for state-of-the-art quality estimating Pr_θ.

The transformer [36], a sequence-to-sequence model built through multi-headed attention layers, has been customized for a number of NLP tasks, as best demonstrated by BERT [10], GPT-2 [29], and a range of notable follow-ups [30, 34, 23]. Conceptually, the attention mechanism works by learning multiple weighted averages per-element of the input sequence. Specifically, this includes three projections of each element's embedding, represented as packed matrices: Q, K, and V. Each projection functions differently, with Q acting as a "query" that is compared against "keys" K and "values" V. The specific mechanism is defined as follows, with d_k representing the dimensionality of each Q and K embedding:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right)V \tag{4}$$

The "multi-headed" aspect of the transformer indicates that the self-attention mechanism is applied multiple times per-layer, per-element of the sequence. These multiple heads are then recombined through a feed-forward layer:

$$\begin{aligned} \text{MultiHead}(X, Y) &= [h_1; \dots; h_k]W^{(4)} \\ \text{where } h_i &= \text{Attention}\left(XW_i^{(1)}, YW_i^{(2)}, YW_i^{(3)}\right) \end{aligned} \tag{5}$$

The transformer model presented by Vaswani et al. [36] use the attention attention mechanism in three different ways. Within the encoder stack, which processes the input sequence in their proposed sequence-to-sequence model, the K, Q, and V embeddings all come from the same sequence of tokens. This is referred

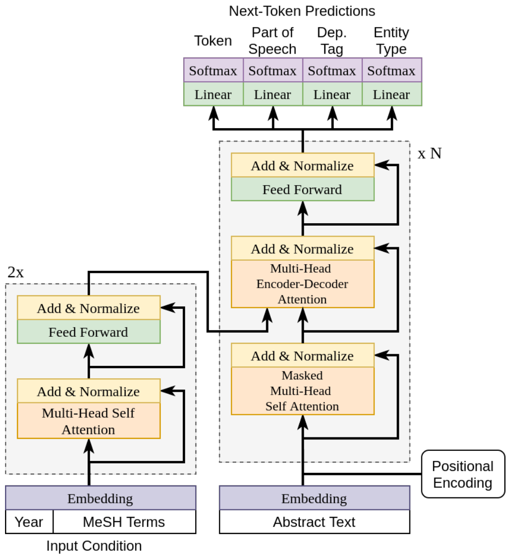


Figure 1: Abstract Generator Model.

to all "self attention." In the decoder stack, the part of the model that uses the encoder output to generate a new sequence, these embedding matrices are masked during the attention function such that the output embedding for position i can only depend on prior elements. This is called "masked self attention". Following this operation, each decoder embedding is attended with all of the encoder embeddings. Specifically, Q values are derived from the decoder, while K and V values depend on the encoder. We refer to this operation as "Encoder-Decoder Attention." Note that BERT [38] uses only the encoder self-attention layers, while GPT-2 [29] uses the decoder's masked self-attention layers. The work presented here uses all three.

The multi-head components are combined with a feed-forward operation, denoted FF, that projects the concatenated embedding into a larger dimensionality, applies the ReLU activation function, and then reduces back to the set embedding rank:

$$\mathrm{FF}(X) = \max(0, XW)W' \tag{6}$$

Then, combined with a learned layer-wise normalization, these components combine to form encoder and decoder blocks. Omitting the standard dropout between each operation, the encoder block is defined as:

$$\begin{aligned} \mathcal{E}(X) &= \mathrm{LayerNorm}(\mathrm{FF}(\alpha) + \alpha) \\ \alpha &= \mathrm{LayerNorm}(\mathrm{MultiHead}(X, X) + X) \end{aligned} \tag{7}$$

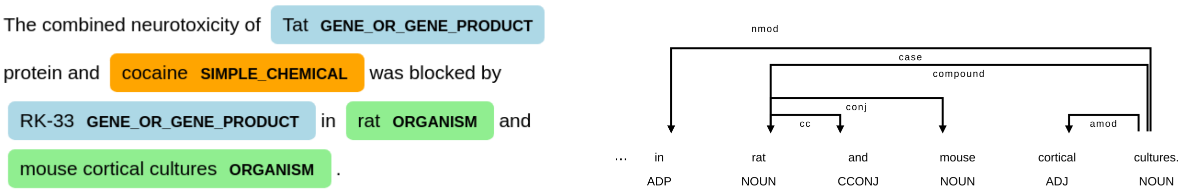
while the decoder block is defined as:

$$\begin{aligned} \mathcal{D}(X, Y) &= \mathrm{LayerNorm}(\mathrm{FF}(\alpha) + \alpha) \\ \alpha &= \mathrm{LayerNorm}(\mathrm{MultiHead}(\beta, Y) + \beta) \\ \beta &= \mathrm{LayerNorm}(\mathrm{MultiHead}(X, X) + X) \end{aligned} \tag{8}$$

3 Multi-Conditional Language Model

The CBAG model follows the transformer architecture [36] with a shallow "condition" encoder, and a deep "language model" decoder. This model is depicted in Figure 1. The condition is specified as a set of embeddings that enable a high degree of control. To capture information that is particular to language within biomedical domain, we add terms in our objective representing not only elements of the textual sequence, but also the part-of-speech, dependency tags, and entity class labels associated with each textual element. For each class of prediction, we minimize the sum of negative log likelihood:

$$\mathcal{L}(t, p, d, e, c) = L_T(t, t, c) + L_P(p, t, c) + L_D(d, t, c) + L_E(e, t, c) \tag{9}$$


The combined neurotoxicity of
Tat GENE_OR_GENE_PRODUCT
protein and
cocaine SIMPLE_CHEMICAL
was blocked by
RK-33 GENE_OR_GENE_PRODUCT
in
rat ORGANISM
and
mouse cortical cultures ORGANISM
.
nmod
case
compound
conj
cc
amod
... in rat and mouse cortical cultures.
ADP NOUN CCONJ NOUN ADJ NOUN

(a) Typed entity recognition.

(b) Dependency tags and parts of speech.

Figure 2: Annotations provided by ScispaCy "BIONLP13CG."

where $t = t_1, \ldots, t_n$ are the set of ground-truth textual elements, each with associated $p_i \in p$ part-of-speech tags, $d_i \in d$ dependency labels, $e_i \in e$ entity labels. The term $c = c_1, \ldots, c_m$ indicates the set of conditions associated with t, and captures information such as metadata keywords and the publication year of the ground truth elements. Each term of (9) follows the form of:

$$L_{[\cdot]}(\ell, t, c) = \sum_{i=1}^{n} -p_{\ell_i}^{(i)} + \log\left(\sum_{j \neq i} \exp\left(p_j^{(i)}\right)\right) \qquad (10)$$
$$\text{where } p^{(i)} = \text{softmax}\left(\mathcal{H}\left(\{t_1, \ldots, t_{i-1}\}, c\right) W_{[\cdot]}\right)$$

where the symbol $[\cdot]$ is replaced by T, P, D, or E for each classification objective. The sequence ℓ indicate the ground-truth labels associated with each element of t with respect to the particular classification task. Additionally, $\mathcal{H}(t, c)$ is the proposed transformer model, which accepts all text elements $\{t_1, \ldots, t_{i-1}\}$ and c in order to produce an encoding for t_i. This model is defined as:

$$\begin{aligned} \mathcal{H}(t, c) &= D_d \\ D_{i+1} &= \mathcal{D}(D_i, E_e) \text{ and } D_0 = t + \text{PE} \\ E_{i+1} &= \mathcal{E}(E_i) \text{ and } E_0 = c \end{aligned} \qquad (11)$$

Here, PE references the positional encoding defined by the sinusoidal function presented in [36]. Each input element of t and c is first assigned an input encoding and put through their respective stacks of encoder and decoder layers. Input encodings are provided by an embedding table that begins randomly initialized. We determine textual elements through the unigram word-part tokenizer [18], and contextual elements consist of a learned embedding per-publication year, as well as embeddings for each Medical Subject Heading (MeSH term). These input factors are described in father detail in Section 4.

Hyperparameters. We selected hyperparameters similar to the GPT-2 "medium" model. This includes an embedding dimensionality of $d_k = 1,024$, $k = 16$ attention heads per multi-headed attention layer, $e = 2$ encoder blocks, $d = 16$ decoder blocks, a fully-connected size of 3,072, and an inner-block dropout rate of 0.1. We additionally use a max sequence length of $n = 128$. Our set of initial embeddings contains 16,000 text tokens, 48,133 MeSH headings, and 230 year embeddings.

Optimization. We minimize $\mathcal{L}$ using the large-batch optimizer LAMB [42] across 40 Nvidia V100 GPUs using an effective batch size of 480. We selected a learning rate of 0.001, with a 500-batch linear warm up. We check pointed the model each epoch after viewing 5% of the training data (about 700,000 abstracts). Note that each time an abstract is viewed, we select from it a different training window. We trained this model for 72 hours using PyTorch Lightning [12] to aid in the distribution and check pointing.

4 Data Preparation

In order to train the model described above, we collect training samples (t, c) from the set of publicly available biomedical abstracts provided in the MEDLINE database. This dataset contains publication dates, author-supplied MeSH terms, titles, and abstracts for mote than 30-million citations. We filter for documents that were originally published in English, as well as documents that contain at least one non-title sentence. Documents without metadata keywords are allowed. We split the remaining 20-million abstracts into a training and test set following a 70-30 split.

Within the domain of biomedical text mining, there are relatively few annotated training sources [15, 26]. To endow the CBAG model with biomedical-domain knowledge, we annotate the entire MEDLINE training set using an NLP model trained on a smaller annotated training set. Because we leverage patterns mined from a small human-annotated dataset to gain broader insights across a vast unstructured dataset, we refer to our overall approach as semi-supervised. The ScispaCy model [24] trained on the "BIONLP13CG" BioCreative dataset [15] provides our biomedical NLP model. This model was selected because it produces the widest range of entity labels when performing named entity recognition, which consist of: cancer, organ, tissue, organism, cell, amino acid, gene or gene product, simple chemical, anatomical system, immaterial anatomical entity, multi-tissue structure, developing anatomical structure, organism subdivision, and cellular component. We add a class corresponding to "not an entity" as well.

Using the ScispaCy model and a cluster of 100 machines, we quickly identify every token, part-of-speech, dependency tag, and entity label for all 14-million training-set MEDLINE documents. We depict examples of these automatic annotations in Figure 2. However, in order to formulate these textual features for input into the CBAG model, we also leverage the unigram subword regularization method from Kudo et al. [18]. This method learns an efficient tokenization sentences. Each token corresponds to a "chunk" of characters, many of which correspond to subword components. The unigram approach adds a normalization factor wherein the specific tokenization for each word is probabilistic determined from the set of ambiguous subword sequences. These subword sequences, along with special "start of abstract" and "end of abstract" tokens, create input t.

We train the unigram tokenization method on one-million randomly sampled sentences from the training set, specifying a fixed-size vocabulary of 16,000 subword tokens. We additionally lowercase the entire training corpus, and enforce that every character within the sampled training set receive its own token. Using the resulting model, we tokenize the entire training set, and cross reference the subwords with the multi-task labels provided by ScispaCy. This way, each subword token t_i in the training set is associated with a part-of-speech p_i, dependency tag d_i, and entity label e_i.

Next we index each training-set publication years and author-supplied MeSH keywords, which form the condition c. For publication years, we simply identify the earliest year within the training set, 1790, and add an index for each year between then and 2020. We identify over 4-million author-supplied keywords within MEDLINE, which is prohibitively large for our model to capture. We prune any keyword that occurs fewer than ten times, reducing that set to a manageable 48,133. We add each to our excising embedding index, which contains nearly 50,000 total embeddings.

When training, we select a batch of abstracts, and for each abstract we select a window of 128 subword tokens to form t, restricted such that the first token of each window corresponds to the first token of a sentence. In addition, we supply the condition indices c. The sequence of labels ℓ is formulated by shifting the subword token window by one token, such that t_{i-1} is used to predict t_i, p_i, d_i, and e_i. An example of model input and output is depicted in Figure 3.

5 Results

While NLP benchmarks such as GLUE [39] and its biomedical counterpart BLUE [26] help researchers compare performance across a range tasks, we are unaware of a benchmark for the generation of biomedical abstracts. In lieu of such a dataset, we leverage our held-out test-set of Medline abstracts, and a set of traditional NLG metrics [31]. We generate abstracts by providing a title t and condition c from a test-set abstract. We extend t by sampling from the resulting probability distribution over subword tokens $p^{(i)}$ until observing the "end of abstract" special token. The quality of the resulting abstract is quantified for each metric, Bleu [25], METEOR [19], ROUGE-L [22], and CIDEr [37], by comparing each generated sentence against the set of "reference" sentences comprising the corresponding human-written abstract.

To add context to our reported performance numbers, we also generate text using OpenAI's recently released 1.5-billion parameter "huge" GPT-2 model [29]. This model has been shown to excel on a number of tasks without modification, inducing as a replacement to traditional knowledge bases [27]. However, as this model was trained to generate language found online, such as in the BooksCorpus and English Wikipedia, it is at a disadvantage when generating domain-specific text. Because GPT-2 does not produce any "end of document" indicator, we generate the same number of subword tokens as present in the human-written

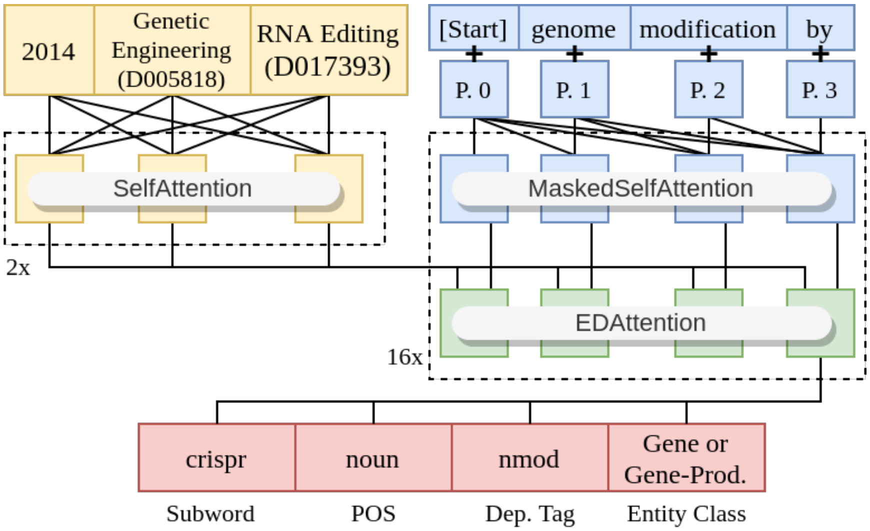


Figure 3: Abstract Generator Example Input.

counterpart, and truncate the potential partial sentence at the end of the abstract.

We present a full abstract from both CBAG and GPT-2 in Table 1. Note, newline characters produced by GPT-2 are replaced with "[\n]" due to space limitations. In this example, we observe that the CBAG model recovers a set of relevant biomedical entities. Unsurprisingly, the model parrots some entities that appear in the title, such as, "micro- and nanopatterned topographical cues," as well as "cellular functions" in this example. However, it is also able to produce more advanced concepts including "multiple imaging modalities," and "multiscale substrates" that do not appear in the title but do appear in the corresponding human-written abstract (not reproduced here for space concerns, but is publicly available). The GPT-2 model does recover some biomedical entities, such as "damaged retinal pathway" and "retinal pigment epithelium," however these keywords are unrelated to the considered document. Other out-of-context entities such as "artificial neural network," "computer screen," and reference to a blog reduce the ability of a human reader to extract any meaningful biomedical information from this text. We find that these example abstracts help motivate the need for domain-specific language models.

Because CBAG is a conditional language model, we explore the range of responses the model can produce given different conditions. In Table 2 we present the first sentence produced by the model for the input "*In this study, we found...*" given different conditions. The results indicate that the condition has a significant impact in the resulting text. When conditioned with the MeSH term for contraceptive agents, the model discusses a patient study on cardiac side-effects. The output conditioned on the pesticide DDT describes fruit and toxicity. The output on gold describes describes gold-nanoparticle sensitivity. These results demonstrate the ability for the CBAG model to learn domain-specific research content provided by various keyword preconditions.

To provide further qualitative comparison between the considered models, we additionally provide a few first-sentences produced given various test-set titles in Table 3. In these sentences, and across the test set, we observe that CBAG produces a number of scientific cliqués. Most clearly, the model captures biomedical turns of phrase such as "in clinical practice." Additionally we observe that it is common for CBAG to produce an entity followed by an abbreviation that it will repeat throughout the text. However, we observe that some abbreviations are not sensible from a human perspective, such as "in-field navigation (oif)." In these cases, the incorrect abbreviation will still be repeated by the model.

Not seen in these first-sentences is a trend for the model to follow major abstract claims with a fictional p-value or sample-size. We find p-values in approximately 10% of abstracts, with a median value of 0.02, and when plotting this distribution of generated p-values we find it matches the expected (and troubling) trend of p-values in real-world science [14].

To provide a more rigorous and scalable analysis of CBAG generations, we turn to a collect of NLP metrics, mentioned above. We use two version of Bleu, one that includes only 1-grams, and one that sums Bleu scores for 1-through-4-grams. We do not apply smoothing or any additional normalization to Bleu

(PMID: 28026780) Hierarchically Micro- and Nanopatterned Topographical Cues for Modulation of Cellular Structure and Function.
CBAG the ability to integrate multiple physiological cues and thereby mediate many cellular functions is critical for many complex life history processes. despite recent advances in high-throughput imaging of biomolecules and their spatiotemporal integration into dynamic structure and function, the precise structural organization and temporal structure of tissue architecture remains poorly understood. here, we present an efficient system for temporally and spatially mapping micro- and nanopatterned topographical cues in organ-specific spatial and temporal properties using multiple imaging modalities. the micro-/nanopatterned geometrical cues can be localized to cell membranes, cells, and proteins. the spatial and temporal dynamics of these local signals are precisely represented by the cross-correlation function, which forms the basis of a geometrical model that accurately provides spatiotemporal information about the spatial location and spatial coordinate of the labels and their functionalities. the model is also capable of correlating the properties of neural cells within their network without affecting the spatial and temporal organization of their spatial features, as well as those of their surrounding tissue. as an example of this model, cell types grown as multilayers are described.
GPT-2 Huge [\n] [\n] The first example of what happens when an artificial neural network'see' complex features on a computer screen is provided by the vision of people with sight loss. The most recent results (e.g., see Krueger's blog) suggest that a person who has a damaged retinal pathway for visual light-scattering called LID is able to'see' objects using low power laser sources. The finding is unexpected because it suggests an adaptive mechanism for recovering limited visual function. The finding has profound implications for understanding how the retina and brain develop, how vision is learned, and how people can potentially regain impaired vision with new retinal implants. [\n] The retinotopic map shows which parts of the retina belong to what parts of the visual field (more details in the video below). [\n] The following images are from a retinotopic map showing a person who has lost some peripheral vision due to an injury in his optic nerve and retinal pigment epithelium. Note that he can make out details when looking toward the left, and when looking toward the right, details become distorted or are obscured.

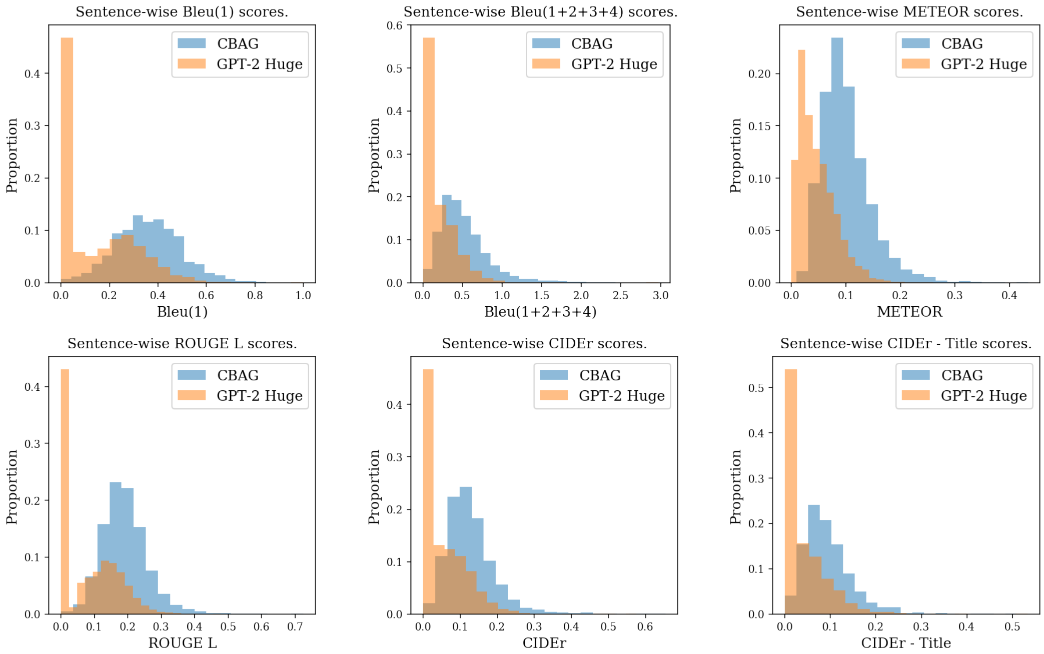
Table 1: Full abstracts generated with respect to the same title.

Condition	Response
D003270: Contraceptive Agents	...that, during a prospective observational period, the patients were aware of the possibility of adverse cardiac events.
D003634: DDT	...that the aromatic (g)-tse, which is often produced in fruit, is potentially useful to suppress green algae as well as pesticide toxicity.
D004042: Unsaturated Dietary Fats	...that vitamin e levels are associated with early childhood health consequences.
D006046: Gold	...that the nanoparticles provide improved sensitivity to gold nanoparticles, and they are sensitive to ag-b interaction rather than ca-a interaction.
D005395: Fish Oils	...that the combination of pinkland and fish oil intakes (ca-like and ca-like) improves the antioxidant effect of yinneria (tricapsa vul) and that can significantly decrease food intake.

Table 2: Differing generations of the same prompt given various MeSH preconditions. We record the first sentence completing the prompt *"In this study, we found..."*

<table>
<tr><th colspan="2">(PMID: 28029317) Laparoscopy to Predict the Result of Primary Cytoreductive Surgery in Patients With Advanced Ovarian Cancer: A Randomized Controlled Trial.</th></tr>
<tr><td>laparoscopic surgery is the standard treatment for patients with advanced ovarian cancer; however, these patients do not receive a standard palliative regimen.</td><td>J Natl Cancer Inst 2008;100:1567–1572. 24. The focus of this review is the effect of apoE4 levels on the risk of poor surgical outcome in patients with advanced ovarian cancer.</td></tr>
<tr><th colspan="2">(PMID: 27993387) Low vitamin D does not predict statin associated muscle symptoms but is associated with transient increases in muscle damage and pain.</th></tr>
<tr><td>in clinical practice, patients with moderate-to-severe hypervitaminosis d present with debilitating side effects related to statin use.</td><td>ow vitamin d does not predict statin associated muscle symptoms but is associated with transient increases in muscle damage and pain.</td></tr>
<tr><th colspan="2">(PMID: 28012718) Skin-Resident Effector Memory CD8+CD284− T Cells Exhibit a Profibrotic Phenotype in Patients with Systemic Sclerosis.</th></tr>
<tr><td>systemic sclerosis (ssc) is an inflammatory disease characterized by the infiltration of t cells into skin and skin surfaces. the presence of autoantibodies can lead to the development of cutaneous t-cell hyperactivity.</td><td>J. Clin. Invest. 117 : 2748-2759; Dilating collagen in chronic neuropathic pain. Arch. Neurol. 63 : 983-989</td></tr>
<tr><th colspan="2">(PMID: 27999935) Laparoscopic sentinel node navigation surgery for early gastric cancer: a prospective multicenter trial.</th></tr>
<tr><td>to compare the feasibility and safety of laparoscopic sentinel node navigation surgery with that of conventional in-field navigation (oif) surgery in the treatment of early gastric cancer (egc).</td><td>Patel S et al. (2003) Age associated factors associated with false-positive result of prognostic biomarkers in prostate and breast cancer.</td></tr>
</table>

Table 3: CBAG (left) compared to GPT-2 “huge” with 1.5B parameters (right). Both systems are given the same title as a prompt. CBAG receives metadata. Results truncated for space.


Sentence-wise Bleu(1) scores.

Sentence-wise Bleu(1+2+3+4) scores.

Sentence-wise METEOR scores.

Sentence-wise ROUGE L scores.

Sentence-wise CIDEr scores.

Sentence-wise CIDEr - Title scores.

Figure 4: Score distributions per-sentence comparing GPT-2 Huge with CBAG.

scores in an effort to reduce unnecessary hyperparameters. Furthermore, we present two versions of CIDEr. While both use a sub-sample of training-set abstracts to approximate n-gram document frequency, we also want to determine whether the generated text can produce uncommon n-grams that were not supplied in the title. Our "CIDER-Title" metric sets the weight of any n-gram that appeared in the title to zero. The sentence-wise score distribution for all metrics for a sample of test-set abstracts are depicted in Figure 4, including both scores for CBAG and GPT-2 generations. Note, these histograms are scaled such that all bars for a particular model sum to one.

We observe that about half of the sentences produced by GPT-2 contain very little content. As seen in Table 3, we see many of these sentences appear to be in the style of citations, including page numbers and titles. Therefore, sentences such as "J Natl Cancer Inst 2008;100:1567–1572. 24." are unlikely to recall many relevant n-grams. Other examples, such as the full GPT-generated abstract shown above, seem to discuss scientific findings from the perspective of an online news outlet covering the new research. While the CBAG generations are imperfect, they do score higher, on average, across all considered metrics. In the case of ROUGE-L, which measures the ability for generated sentences to recall long sub sequences of text, that many biomedical cliqués are likely easy for CBAG to predict, such as "the study examined the" or "we conclude that the." Our higher METEOR scores, which indicates the ability to recall n-grams in the same order as found in a reference sentence, are also effected by these common sequences. However, the "CIDEr-Title" metric explicitly decreases the weight of these common n-grams, while only considering text that could not be identified trivially. Our improved performance in this measure, when seen in the context of our overall improvement, demonstrates the ability for CBAG to produce more relevant and nontrivial biomedical text than the baseline.

6 Related Work

SciBert [5] achieves state-of-the-art performance across a range of scientific NLP benchmarks by retraining the WordPeice tokenizer [40], and a BERT model [10] on 1.14-million papers collected by semantic scholar. Beltagy et al. demonstrate that by performing unsupervised pre-training on this scientific dataset, they are able to improve performance over the standard BERT-pre-trained weights on their ultimate fine-tuned models for entity recognition, PICO extraction, text classification, relation classification, and dependency parsing. These finding make the case that scientific text is sufficiently dissimilar from that found in general language to require custom models.

BioBert [20] follows the same pattern as SciBert, but pre-trains on the biomedical texts supplied by MEDLINE and PubMedCentral. As opposed to SciBert, this method does not replace the general-language training data supplied by English Wikipedia and BooksCorpus, and instead appends both biomedical text databases. Lee et al. explore the resulting fine-tuned performance across a large range of small biomedical NLP tasks, and find mixed results. We interpret these results to indicate the importance of finding training data that is not only sufficiently large, but also relevant to the task at hand.
Wang et al. [38] explore the capacity for a BERT model to effectively function as a Markov random field language model. This technique takes advantage of the masked pre-training used in the base BERT model to predict unknown tokens. This approach also departs from the traditional language model described here as every sequence element determines the probability of every other element. Generation is performed by iterative freezing highest-probability elements from within a fixed-length sequence of initially free variables.
Ctrl [17] is a conditional language generation method that extends GPT by including "control codes" that prefix the sequence of text elements. For instance, each website represented in the training data is represented by a code, and as a result generated text can switch styles based on these prefixes. Additionally, various model functions, such as question answering, are learned via generation with various codes. As a result, prefixing questions with the respective code results in a higher probability assigned to relevant answers. Furthermore, this work includes some multi-code prefixes, such as "Rating 5.0" or "Sentence Title" to further condition the generated result. While the CTRL model is the most similar to the method presented here, it has some key differences. Firstly, the CTRL model uses prefix tokens to condition generated text, while we apply a shallow transformer-encoder stack. As a result, the CTRL approach is limited in that training requires a strict set of codes, or a small set of enumerable code-pairs. In contrast, the CBAG approach allows the method proposed here to accept arbitrary-length sequences of keywords as a condition.

7 Future Challenges and Ethical Considerations

Many readers have likely heard of "hoax" paper generators similar to Scigen [33]. This particular project generates computer science full-text articles by randomly sampling from a context-free grammar, and has produced publications actually accepted by some venues. This 15-year-old system, however, is incapable of fooling but the least-observant of gate keepers. However, high quality text generation introduces NLP to a range of challenges currently posed by "deepfake" images. These problematic pictures permeate the zeitgeist and stir a response reaching further than computer science [11], extending into law [7], culture [2], and philosophy [13]. Meanwhile, misinformation spread by human actors online already cascades throughout social network echo chambers at an alarming rate [9]. One needs very little imagination to conceive of ways that the automatic generation of "pseudo-science" online could lead to public distrust of the scientific community.

OpenAI is forming partnerships between computer-science and the social sciences in order to understand these implications in society [21]. One major challenge they note is a distinct lack of "correctness" measures for text generation. In completing this work, we find that some correctness measures do exist, such as the SPICE metric to judge image caption correctness [3]. Unfortunately, this technique does not scale well to large knowledge bases as it requires the graph of predicate arguments induced by reference sentences. Not only are there a lack of methods to extract arguments from text, but we need to find new algorithms for quantifying correctness for large graphs induced by all of biomedical science.

Despite the potential for abuse, we designed CBAG with our own vision toward enabling human-understandable hypothesis generation systems. For instance, our model architecture could be conditioned on more generalized forms of existing biomedical knowledge, such as semantic graph embeddings, in order to produce textual descriptions of plausible future research directions. These explanations could potentially persuade domain scientists to pursue new research directions, as similar systems have already done [1, 4]. However, these systems require specialized analysis and introduce new cognitive burdens for scientists to understand and act on their outputs. If similar hypothesis generation systems instead could produce human-readable arguments, then we could better utilize the wealth of publicly available information, improve the productivity of biomedical researchers, and ultimately find new treatments and cures for people worldwide.

8 Conclusions

We present the Conditioned Biomedical Abstract Generation (CBAG) model for understanding scientific abstracts. We train this model using publicly available biomedical data provide through MEDLINE to predict text that is conditioned on publication year and arbitrary sets of author-supplied keywords. This model leverages the transformer architecture [36], featuring a shallow condition encoder, as well as a deep language model decoder. Using CBAG, and a range of natural language generation metrics [31], we demonstrate the need for such a domain-specialized model, as opposed to a larger more general model like GPT-2.

We anticipate that conditioned language generation can be used to build new applications in the biomedical domain, such as a hypothesis generation system that produces textual descriptions of proposed new research directions. To do so, the conditional aspect of the CBAG model will likely be a necessity. However, we also acknowledge the ethical considerations behind the proliferation of convincing scientific language generation models. We provide the pre-trained model, over 13,000 generated abstracts, and all necessary training and evaluation code to aid in exploration and reproduceability.

References

[1] Marina Aksenova, Justin Sybrandt, Biyun Cui, Vitali Sikirzhytski, Hao Ji, Diana Odhiambo, Matthew D Lucius, Jill R Turner, Eugenia Broude, Edsel Peña, et al. Inhibition of the dead box rna helicase 3 prevents hiv-1 tat and cocaine-induced neurotoxicity by targeting microglia activation. *Journal of Neuroimmune Pharmacology*, pages 1–15, 2019.

[2] Oxford Analytica. 'deepfakes' could irreparably damage public trust. *Emerald Expert Briefings*, (oxan-db).

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[4] Nadine Bakkar, Tina Kovalik, Ileana Lorenzini, Scott Spangler, Alix Lacoste, Kyle Sponaugle, Philip Ferrante, Elenee Argentinis, Rita Sattler, and Robert Bowser. Artificial intelligence in neurodegenerative disease research: use of ibm watson to identify additional rna-binding proteins altered in amyotrophic lateral sclerosis. *Acta neuropathologica*, 135(2):227–247, 2018.

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[7] Marc Jonathan Blitz. Lies, line drawing, and deep fake news. *Okla. L. Rev.*, 71:59, 2018.

[8] Peter Bruza and Marc Weeber. *Literature-based discovery.* Springer Science & Business Media, 2008.

[9] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[11] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.

[12] W.A. et al. Falcon. Pytorch lightning. `https://github.com/PytorchLightning/pytorch-lightning`, 2019.

[13] Luciano Floridi. Artificial intelligence, deepfakes and a future of ectypes. *Philosophy & Technology*, 31(3):317–321, 2018.

[14] Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. The extent and consequences of p-hacking in science. *PLoS biology*, 13(3), 2015.

[15] Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of biocreative: critical assessment of information extraction for biology, 2005.

[16] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017.

[17] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[18] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[19] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231, 2007.

[20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[21] Claire Leibowicz, Steven Adler, and Peter Eckersley. When is it appropriate to publish high-stakes ai research. *Partnership on AI blog post*, 2019.

[22] Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. An information-theoretic approach to automatic evaluation of summaries. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 463–470. Association for Computational Linguistics, 2006.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[24] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*, 2019.

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[26] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, 2019.

[27] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[29] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

[31] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799, 2017.

[32] Scott Spangler. *Accelerating Discovery: Mining Unstructured Information for Hypothesis Generation*. Chapman and Hall/CRC, 2015.

[33] Jeremy Stribling, Max Krohn, and Dan Aguayo. Scigen-an automatic cs paper generator, 2005.

[34] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[35] Justin Sybrandt, Michael Shtutman, and Ilya Safro. Moliere: Automatic biomedical hypothesis generation system. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1633–1642, New York, NY, USA, 2017. ACM.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[37] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[38] Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*, 2019.

[39] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[40] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[41] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[42] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 1(5), 2019.