M5 MBR - January 2021

Executive Summary

Our first MBR in November 2020 focused on establishing basic model training capabilities and performance baselines. For this MBR we have continued to prioritize developing capabilities as opposed to optimizing end-to-end results on partner team tasks. In the past eight weeks our efforts have focused on (1) transitioning our hardware infrastructure to P4 instances to produce a 4.17x reduction in the cost to train a GPT-2-scale (1.5B parameter) model at a one-day cadence, (2) demonstrating the ability to train 10B parameter models, (3) developing multi-lingual and multi-modal training to support structured data (e.g. length, width, size) and ASINS from US, DE, and UAE, (4) developing model quantization capabilities which produce up to a 2x performance improvement in CPU model inference with less than a 0.72% reduction on downstream task accuracy, (5) expanding the coverage and freshness of our data pipeline to include 7B images, review text, and refreshes for text and structured data at a one-day cadence, and (6) using fine-tuning to support the requirements of our partner team applications. With the exception of image training, we are now feature complete according to our Q1 2021 roadmap two months ahead of schedule. These improvements leave us free to focus the remainder of our efforts in Q1 2021 on improving end-to-end results in our partner team engagements.

We have grown our group by 3 AS and 5 SDE (though we are still lacking computer vision expertise). We have also increased our partner team engagements to include an additional three teams beyond our founding members (The Cost Is Right, Fraud Detection, and KPEX), and finalized 2021 launch goals with the Catalog, Semantic Search, QU, and Sponsored Products teams.

Performance

Training performance is an important contributor to the success of the M5 program. Improvements in performance correspond to an increase in experimental agility, and an ability to stay up to date with the Catalog data we rely on. We spent the last month focusing on improvements in performance through changes in infrastructure. We finished migrating our P3 training infrastructure to a cluster of 48 p4dn.24xlarge instances and integrated our training code with AWS Batch to support job scheduling.

The introduction of a job scheduler allows us to train models with no downtime between trainings making more cost-efficient use of our infrastructure (we hold annual leases instead of paying by hour). Our new P4 infrastructure also provides a 4.17x reduction in the cost to train a GPT-2-scale (1.5B) parameter model at a one-day cadence from $10K to $2.4K. This is more than twice the 2x that we originally projected: P4 instances offer twice the peak FLOPs, four times the memory bandwidth, and are only 5% more expensive than P3 instances (we anticipated double). This infrastructure also increases the largest model we can train without having to use model parallel techniques from 12-14B to 15-19B parameters (the larger memory provided by P4 instances leads to an increase in the largest model that can fit on a single GPU). We performed a proof of concept and measured the cost to train a 10B parameter model on our P4 infrastructure at a one-day cadence. The result was just less than $19K. While this is expensive, the cost is dominated by communication inefficiency. We expect this to improve as our P4 infrastructure optimization matures, leading to as much as a 40% reduction in cost (see Appendix A for details).

Data Pipeline

The M5 project is based on the idea that more data, updated more frequently, leads directly to higher quality models. Our current models are trained using text data consisting of 3B ASINs from a static catalog snapshot which was generated in November 2020. In the past month we expanded our static dataset to include images for 7B ASINS from UIFC/Catalog team and reviews along with attributes such as verified tags and ratings. We have also completed the infrastructure necessary to update ASINs at a daily cadence with one-hour retrieval latency. In the coming months we will use this new data to improve our ability to provide high quality representations for Amazon entities.

Model Training / Fine-Tuning

At the end of 2020, the M5 team had the ability to train GPT-2-scale text-only models using ASIN text data consisting of title and description. In the last month we have added several new capabilities: (1) the use of bullet points as an additional source of text data, (2) the use of structured attribute/value pairs, and (3) the use of text data from multiple languages: EN, DE, and AR. The choice to focus on structured attribute/value pairs as a first source of multi-modal data differs from our original goal of using image data. The pivot was motivated by an observation made by the Catalog team that structured data produces more useful signal than image data for the duplicate detection task. Supporting these new capabilities improves our ability to produce high-quality representations, but also brings new model design challenges such as an increase in vocabulary size, how to represent structured data, and how to sample training examples from corpora in which different languages are non-uniformly represented (see Appendix B for details).

Over the past month we expanded M5 model training into a two-step process: (1) models are pre-trained on the application-agnostic masked word prediction (MLM) task (the model is presented with text or structured data with some words or word pieces removed, and the model is asked to predict the missing value), and (2) models are fine-tuned using data provided by partner teams. Generally speaking, fine-tuning simply involves continuing to train a model using new training examples and an application-specific loss function.

A common feature of our collaborations with Catalog and Semantic Search is that their fine-tuning tasks are *binary: t*he catalog team's task involves comparing two ASINs and judging whether or not they are duplicates, and the Semantic Search task involves comparing two ASINs and judging whether they were co-purchased for the same query. Supporting binary tasks requires us to modify the structure of our pre-trained model before fine-tuning: we currently use a *two-tower* approach in which we connect two copies of the pre-trained model with additional layers, and then fine-tune both of those copies in parallel.

The use of fine-tuning produced mixed results on down-stream tasks (see Appendix C for details). Our best performing fine-tuned models demonstrated an accuracy of 86% on the duplicate detection task (+6% over last MBR) and 80% recall on the semantic search task (-2% over last MBR). We are beginning to explore an alternate fine-tuning approach where we present the model with multiple ASINs separated by a special token. We expect this to lead to improvements in these metrics. We are also looking into automatic processes for generating hard negative fine-tuning examples for the semantic search task, as we suspect that a lack of difficult fine-tuning data was partially responsible for the drop in performance compared to our previous MBR, since the model quickly converges to high accuracy (96%) on the fine-tuning task.

Model Quantization

Quantization, distillation, compression, and pruning are the four dominant post-training approaches for creating smaller and faster deep learning models. These steps are crucial to ensuring that multi-billion parameter models can meet production SLAs for model inference (see Appendix D for details). Quantization replaces high-precision model weights (e.g. float32) with lower precision alternatives (e.g. int8), distillation attempts to train a smaller model using the outputs of the original model to provide ground truth labels, compression attempts to remove layers from a network in a way that does not affect model quality, and pruning attempts the same but at the level of individual neurons. We spent the last month focusing on developing quantization techniques as they are both easy to implement and can be applied regardless of underlying model architecture. While quantization alone is insufficient for meeting production SLAs, it is an important first step.

Using int8 values to represent model weights (as opposed to the standard float16) both reduces memory overhead by 2x and improves runtime performance on CPU targets (by as much as 1.33x on c5 instances for inputs with sequence length 512, 2x for inputs with sequence length 128). The improvement in performance is primarily due to our ability to take advantage of special purpose CPU instructions which are designed specifically for vector arithmetic on int8 values (integer operations are generally faster than floating-point operations). Importantly, quantization has a negligible effect on application performance. We quantized a 500MM parameter text model and observed only a 0.34% loss in accuracy on the MLM prediction task, and a 0.72% AUC loss on the offline duplicate detection task.

Partner Engagements

In the past month we completed our first round-trip experiments with the Catalog team. The Catalog team uses M5 ASIN representations to help train a multi-modal ensemble model which is used to determine whether two ASINs are duplicates. The model combines M5 representations with other features such as image data and product type information. We provided the Catalog team with two sets of representations: one from a 500MM parameter fine-tuned text-only model, and one from a 500MM parameter fine-tuned multi-modal model (see Appendix E for details). The text-only representations resulted in no meaningful changes in accuracy (showing 1% or less decrease in AUC for Consumables, HardLines, Media, and SoftLines) and the multi-modal representations showed similar results (1% or less increase in AUC for Consumables, and HardLines, and 1% or less decrease in AUC for Media and SoftLines).

In the past month we also completed several additional round-trip experiments with the Semantic Search team. The Semantic Search team uses M5 ASIN representations to map the Amazon catalog in to a high-dimensional search space. They then use a machine learned model to map queries into that same space and k-nearest-neighbor search to return the top-k semantic search results which are nearest to the query. The results are evaluated in terms of the number of exact matches they contain as determined by human evaluations. In the best case we observed results which were just slightly worse than the reference implementation (70.9% vs 71.8%). As discussed above, this suggests that the fine-tuning dataset we use for this task could be improved.

We expect both of these results to continue to improve going forward as we train our models to convergence (for experimental agility our multi-lingual and multi-modal models are currently trained for ~2.5x fewer steps than our text-only models), identify the best model architectures, configurations, and hyper-parameters, expand the set of structured attribute fields that we train on, and include image and review data. We are also working with both partner teams to automate pieces of our experimental pipelines in order to improve the throughput and latency of round-trip experiments.

We have still yet to begin engagements with the Sponsored Products (SP) team. Recall that we chose to defer this engagement in Q4 2020 due to an early lack of support for fine-tuning (which the SP team believed to be a necessary condition for generating meaningful results). We will be revisiting this engagement now that we have this capability. Several other teams--The Cost Is Right, Fraud Detection, and KPEX--have also expressed interest in using M5 representations. We have provided this data and are waiting on initial results. Appendix F contains a listing of our shared 2021 partner team goals.

Hiring

In the past month we increased our headcount by three AS: one new L4 hire, one new L5 hire, and one L6 from Catalog who will be embedded in the M5 team. We also hired two L4 SDEs, two L5 SDEs, and made an offer to a sixth L5 SDE. This puts us mostly on track for our 2021 Q1 hiring goals. The primary area of expertise which we currently lack is in computer vision. We are currently seeking candidates who fit this profile.

Hits

- We have completed the transition from P3 to P4 instances ahead of schedule and observed a reduction in TCO which is more than twice what we originally projected.
- We have completed an early proof of concept which demonstrates the ability to scale to 10B parameter models. While this is beyond the scale that we intend to train for the foreseeable future, the lessons learned have helped us reduce the cost of training 1.5B parameter models at a one-day cadence from $10K to $2.4K.
- We have built the capability to train multi-lingual models two months ahead of schedule. This feature completion enables us to focus the rest of Q1 on improving the quality of round-trip experiments with partner teams.

Misses

- The round-trip time required to complete partner team evaluations is still longer than we would like. We have made improvements in automation, but manual intervention is still responsible for days of experimental overhead. Of particular note is the time required for the Semantic Search team to generate hand labeled judgements for semantic search results. These are produced by a third-party vendor (Toloka) with unpredictable turnaround times of up to one week or longer.
- We don’t yet have a good fine-tuning dataset for the semantic search task. While we’ve converged on ASIN co-purchase data, the first version of the dataset was too sparse, and the second version does not contain enough hard negatives. We will make automating the process of generating hard negative examples a priority going forward.

Discussion Points

- What can we do to further reduce the roundtrip time on experiments with partner teams? Are there fundamental reasons (e.g. scientific or architectural) why these processes can’t be fully automated? Or is this a resourcing issue which can be solved with additional headcount?
- Removing the dependence on Toloka evaluations from our experimental loop with Semantic Search requires a different strategy for generating ESCI judgements. What can we do to accelerate the development of this capability? Should we partner with the Sentinel Team (which is currently developing ESCI classification models) to tune their models for this scenario?

Appendix A: Performance Evaluation

The table below summarizes the time required to train 1.5B and 10B models using P4 instances. Both models use a global batch size of 8192, and training iterations are split 9:1 between sequence lengths of 128 and 512. All configurations use 16 instances. Activation checkpointing, micro-batch size, and optimization level are parameters which control the behavior and performance of the DeepSpeed training infrastructure. Profiling shows that the 10B parameter cases demonstrate sub-optimal communication bandwidth utilization (62%) and communication-computation ratio (5.6). To address this, we are exploring techniques for compressing gradients and the use of Herring. Herring is an AWS distributed framework which provides alternate communication collective primitives with better communication performance.

Model Size (parameters)	Sequence Length	Micro-batch Size	Zero Optimization Level	Activation Checkpointing	Throughput (samples/s)	Iterations	Training Time (days)
1.5B	128	64	2	no	12300	112500	0.91
	512	4	1	no	4703	12500	0.33
10B	128	64	2	yes	1688	112500	6.51
	512	8	2	yes	345	12500	3.29

Appendix B: Training Evaluation

Multi-lingual Training: Training a multi-lingual model introduces two new complications: an increase in vocabulary size and the possibility that languages with fewer examples will be underrepresented during training. The table below summarizes MLM accuracy results for a 500MM parameter model. We varied vocabulary size from 32K to 256K and the parameter α was used to control language sampling weight (lower values of α correspond to more samples taken from under-represented languages). We observe that results are insensitive to vocabulary size, and that oversampling underrepresented languages (e.g. Arabic) can improve MLM results for that language, but beyond a certain threshold (α=0.7) can negatively affects other languages (e.g. English).

Vocab Size	Dataset	MLM accuracy(%) Seq 128 Training	MLM accuracy(%) Seq 512 Training	MLM accuracy(%) for EN. (Seq 512)	MLM accuracy(%) for DE. (Seq 512)	MLM accuracy(%) for AR. (Seq 512)
32K	Locale Weighted	89.1	89.3	**91.8**	89.4	87.1
	Multinomial/ α = 0.7	89.6	89.6	91.7	**89.4**	88.2
	Multinomial/ α = 0.3	**89.9**	**90**	89.8	88.9	**89.8**
64K	Locale Weighted	88.1	88.2	91.6	**88.8**	84.7
	Multinomial/ α = 0.7	88.6	88.6	**91.6**	88.7	85.9
	Multinomial/ α = 0.3	**89**	**88.9**	91.3	88.2	**87.6**
128K	Locale Weighted	87.6	87.6	91.6	88.4	83.3
	Multinomial/ α = 0.7	88.1	88	**91.6**	**88.7**	85.9
	Multinomial/ α = 0.3	**88.5**	**88.4**	91.3	87.9	**86.3**
256K	Locale Weighted	87.4	87.4	91.7	**88.3**	82.6
	Multinomial/ α = 0.7	87.9	87.8	**91.7**	88.2	83.9
	Multinomial/ α = 0.3	**88.4**	**88.1**	91.4	87.8	**85.7**

Multi-modal Training: Training a multi-modal model based on structured attribute/value pairs introduces new complications as well: how best to delimit pairs, and whether to represent values using sub-word tokenization or as enumerations. The table below summarizes value prediction accuracy for a 500MM parameter model. We varied whether pairs were delimited using special characters or presented as a text stream, and whether values were masked at the sub- or whole-word granularity. With respect to value prediction accuracy, we observe the best results when using no special tokens and sub- word masking. In this configuration, the value-prediction task effectively devolves to the MLM prediction task. Interestingly, this result does not extend to downstream tasks (see Appendix C). There we observe the opposite, indicating that a data representation scheme which communicates the structure of the training data improves a model's ability to learn representations which are useful for duplicate detection and semantic embedding.

Special Tokens	Word Masking	Value Prediction Accuracy (%)
none	Sub-Word	**90.6**
	Whole-Word	89.7
field name	Sub-Word	87.3
	Whole-Word	87.3

Appendix C: Fine-Tuning Evaluation

Duplicate Detection: The table below compares performance on the duplicate detection task using 500MM parameter fine-tuned two-tower models. The models are split into three types: text-only (top), multi-modal (middle), and multi-lingual (bottom). All three types produce results within 1% of one another. This is likely due to us having stopped pre-training before convergence for our multi-lingual and multi-modal models in the interest of sampling more datapoints. Given additional pre-training, we expect that the multi-modal model would have performed better. That said, the fact that the multi-lingual model did not under-perform the alternatives is encouraging. This suggests that our models have more than enough capacity to represent multiple languages without a negative effect on downstream task performance.

Language	Model Size	Notes	Pre-Training Data	Fine-Tuning Data	Test AUC
EN	500MM	-	Title/Desc	Title/Desc	84.9
EN	500MM	-	Title/Desc	Title/Bullet	84.7
EN	500MM	-	Title/Desc	Title/Structured	85.7
EN	1.5B	-	Title/Desc	Title/Desc	85.8
EN	1.5B	-	Title/Desc	Title/Structured	**86**
EN	500MM	No special tokens/ Sub-word masking	Title/Desc/ Structured	Title/Bullet/ Structured	85.2
EN	500MM	No special tokens/ Whole-word masking	Title/Desc/ Structured	Title/Bullet/ Structured	85.1
EN	500MM	Special tokens/ Sub-word masking	Title/Desc/ Structured	Title/Bullet/ Structured	**85.7**
EN	500MM	Special tokens/ Whole-word masking	Title/Desc/ Structured	Title/Bullet/ Structured	85.5
EN/DE/UAE	500MM	32K Voc	Title/Desc/Bullet	Title/Desc/Bullet	**85.3**
EN/DE/UAE	500MM	32K Voc / α = .3	Title/Desc/Bullet	Title/Desc/Bullet	85.2
EN/DE/UAE	500MM	256K Voc	Title/Desc/Bullet	Title/Desc/Bullet	**85.3**
EN/DE/UAE	500MM	256K Voc / α = .3	Title/Desc/Bullet	Title/Desc/Bullet	85.1

Semantic Search: The table below compares performance on the semantic matching task 500MM parameter models. The models are either text-only or multi-modal, and exclusively pre-trained or pre-trained and then fine-tuned. All models produce representations with arity 768 or 1024. Where noted (e.g. 768->256) we have scaled these representations down to 256 elements. In all cases, a DSSM model was used to generate query representations of matching arity to the ASIN representations. None of the models we evaluated were able to produce results which matched the use of a DSSM model for generating both query and ASIN representations, not even those which were fine-tuned. We believe this shows that (1) fine-tuning is essential for obtaining good results on this task and (2) our fine-tuning dataset is not currently good enough (e.g. does not contain enough difficult negative examples) to achieve those results.

Embedding Dimension	Model	Pre-Training Data	Fine-Tuning Approach	Fine-Tuning Data	Embedding Generation Data	MAP (Weighted / Unweighted)	Recall@100 (Weighted / Unweighted)
768	DSSM	-	-	-	Title	**0.518/0.465**	**0.878/0.886**
1024	DSSM	-	-	-	Title	0.515/0.467	0.874/0.885
768	Quipus	-	-	-	Title	0.333/0.334	0.651/0.685
768->256	M5	-	-	-	Title	0.438/0.372	0.777/0.763
768	M5	Title/Desc	-	-	Title	0.200/0.229	0.506/0.580
					Title+Desc	0.194/0.194	0.501/0.512
768->256	M5	Title/Desc	-	-	Title	0.463/0.376	0.805/0.775
					Title+Desc	0.464/0.363	0.798/0.750
1024	M5	Title/Desc	-	-	Title	0.152/0.146	0.425/0.480
					Title+Desc	0.192/0.172	0.464/0.427

1024->256	M5	Title/Desc	-	-	Title	0.428/0.335	0.767/0.699
					Title+Desc	0.377/0.304	0.712/0.644
1024	M5	Title/Desc	Two-Tower	Title/Desc	Title	0.117/0.107	0.379/0.349
					Title+Desc	0.119/0.106	0.382/0.349
1024->256	M5	Title/Desc	Two-Tower	Title/Desc	Title	0.323/0.274	0.625/0.555
					Title+Desc	0.315/0.267	0.615/0.546
768	M5	Title/Bullet/ Structured	-	-	Title/Bullet/ Structured	0.329/0.311	0.677/0.679
768->256	M5	Title/Bullet/ Structured	-	-	Title/Bullet/ Structured	**0.471/0.364**	**0.807/0.756**

Appendix D: Quantization Evaluation

Performance Study: The table below measures inference performance for a 500MM parameter model on a broad selection of AWS instances. In these evaluations, we fix the input sequence length to 512 (for real-world workloads, this value would typically be closer to 128), set the number of CPU threads to 16, and utilize all available GPUs when applicable. Production systems typically require inference times of 15ms or less. In the absence of distillation techniques, we are unable to meet this SLA, even with GPU instances. Quantization helps, but by itself is not a solution. We note that we were unable to apply quantization to GPU targets due to current limitations in our PyTorch infrastructure. We will revisit GPU quantization later, with low priority, as we do not want to require partner teams to make a large investment in GPU infrastructure in order to take advantage of M5 models.

Instance Type	Instance Type (number of chips)	Model	Batch Size	Latency (ms)	Throughput (requests/s)	Throughput per price (throughput/$)
c5.24xlarge	CPU	Baseline	1	645.2	2.87	0.01019
		Quantized	1	485.5	3.86	**0.0137**
r5.24xlarge	CPU	Baseline	1	655.9	2.76	0.00517
		Quantized	1	566.5	3.27	0.00613
m5.24xlarge	CPU	Baseline	1	656.1	2.77	0.00939
		Quantized	1	554.6	3.22	0.01092
g4dn.16xlarge	GPU (1)	Baseline	1	187.6	5.35	0.00587
		Baseline	64	162.1	6.45	0.00708
p2.8xlarge	GPU (8)	Baseline	1	445.8	25.36	**0.10042**
		Baseline	64	339.9	53.25	0.21085
p3dn.24xlarge	GPU (8)	Baseline	1	66.8	50.93	0.01463
		Baseline	64	47.6	351.04	0.10081
p4dn.24xlarge	GPU (8)	Baseline	1	59.9	47.63	0.01303
		Baseline	64	10.65	1068.6	**0.29225**

Application Study: The table below summarizes performance on the duplicate detection task both before and after model quantization. In addition to the performance improvements shown above, quantization produces a reduction in memory, and does not meaningfully affect accuracy.

Model	MLM Accuracy	Duplicate Detection AUC	Model Size On Disk (MB)	Model Size in Memory (MB)
500MM Baseline (fp 16)	0.92	0.833	2056.635	2379 (CPU) / 3363 (GPU)
500MM Quantized (int8)	0.916	0.827	1133.264	1450 (CPU)

Appendix E: Partner Evaluation

Duplicate Detection: The table below summarizes changes in AUC on the duplicate detection task measured on four product lines, using representations from two 500MM parameter M5 models: one text-only, the other multi-modal.

Model	Product Line	Test AUC (change)	Test AUC (percent change)
Text-Only	consumables	-0.01003	-1.15365
	hardlines	-0.00354	-0.38806
	media	-0.00354	-0.36633
	softlines	-0.00324	-0.34803
Multi-Modal	consumables	**0.00768**	0.88309
	hardlines	**0.00283**	0.30935
	media	-0.00093	-0.09591
	softlines	-0.00099	-0.10663

Semantic Search: The table below summarizes the performance of M5 models and the Quipus transformer model on the semantic search task. % Exacts measures the number of exact search results as measured by a human auditor for search results generated according to a fixed corpus of test queries. In the best case, M5 models produce near parity, coming within 1% of the reference implementation.

Embedding Dimension	Model	Query Embedding	Pre-Training Data	Embedding Generation Data	%Exacts (Weighted / Unweighted)
768	DSSM	DSSM	-	Title	0.718/0.667
1024	DSSM	DSSM	-	Title	**0.723/0.671**
768	Quipus	DSSM	-	Title	0.708/0.656
768->256	M5	DSSM	-	Title	0.700/0.644
768	M5	DSSM	Title/Desc	Title	0.622/0.579
				Title+Desc	-
768->256	M5	DSSM	Title/Desc	Title	0.709/0.654
				Title+Desc	0.686/0.631
1024	M5	DSSM	Title/Desc	Title	0.570/0.526
				Title+Desc	0.491/0.456
1024->256	M5	DSSM	Title/Desc	Title	0.556/0.508
				Title+Desc	0.566/0.524

Appendix F: 2021 Shared Partner Goals

Kingpin #270262:
Description: Deliver TK submissions based on M5 technologies in top-tier peer-reviewed machine learning and retrieval conferences.
How is this Measured: Peer-reviewed conferences are ranked according to H5-index: the largest number h such that h articles published in the previous five years have at least h citations each. For 2020, the top five machine learning conferences according to this metric are: ICLR, NeurIPs, ICML, AAAI, and Expert Systems with Applications. For retrieval, this includes SIGIR and WSDM.
Why is this Important: Academic publications establish Amazon as a science and thought leader in Machine Learning and associated technologies. This affects Amazon's recruiting efforts by improving our ability to attract top talent, especially new PhD graduates, and establishing ourselves as a thought leader in the scientific community.

Kingpin #270621:
Description: Reduce the dollar cost required to train a 1.5B parameter model at a one day cadence (from ~$10K to TK) using AWS P4 instances, advances in systems engineering, and improved algorithm design.
How is this Measured: $10K/training is measured relative to the M5 team's training capabilities at the end of Q4 2020. Using 48 p3dn.24xlarge instances the M5 team was able to train a 1.5B parameter transformer model on 1B ASINs (~110B tokens) in ~43 hours at a cost of ~$10K per training. At this rate, 87 instances would be able to train a 1.5B parameter model at a one-day cadence.
Why is this Important: Training cadence determines the M5 team's experimental agility and its ability to provide representations to partner teams in a timely fashion. P4 instances offer improved TCO via higher FLOPs and network bandwidth, and improved systems engineering allows for efficient scaling. These technologies allow for a reduction in the time or number of machines required to train a model of a given size, an improvement in the ability to train multiple models in parallel, or the ability to scale to training larger models as necessary.

Kingpin #273858:
Description: AWS is developing a hardware accelerator which targets training for deep learning models (Trainium). Trainium is projected to offer 6-8x TCO improvement over P3 and is scheduled for private preview Q4 2021, with early Q3 access provided to partner teams. The M5 team will build a POC implementation of model training which demonstrates a TK improvement the dollar cost required to train a GPT-3-scale 100B parameter model at a one day cadence and is presented at ReInvent 2021.
How is this Measured: TCO is measured relative to the M5 team's training capabilities at the end of Q4 2020. Using 48 p3dn.24xlarge instances the M5 team is able to train a 1.5B parameters transformer model on 1B ASINs (~110B tokens) in ~43 hours at a cost of ~$10K per training. GPT-3 scale is defined as 100B parameters.
Why is this Important: Developing a proof of concept in advance of general availability will improve the AWS team's ability to harden its compiler infrastructure. A success story at ReInvent will generate early publicity for Trainium which will lead an increase in both public and internal adoption of a technology which Amazon has invested heavily in.

Kingpin #270626:
Description: Use M5Vend to provide distilled M5 models to the UIFC team for vending and fine-tuning. M5Vend is a Python-based toolkit being developed by AWS which supports "import M5.model-style" access to M5 models. M5Vend will provide a set of easy-to-use tools for performing inference, distillation, and fine-tuning in a platform agnostic fashion.
How is this Measured: "Providing a model" is defined as encapsulating an M5 model in a lossless Tensorflow protobuf-style format and using M5Vend on the UIFC side to perform inference and fine-tuning on that model.
Why is this Important: Low-level systems engineering issues such as the use of accelerator kernels and language/framework incompatibilities make it difficult to vend Python-based deep learning models in a platform-agnostic fashion, leading to performance and accuracy regressions. Abstracting these issues behind an easy-to-use Python-based API will decrease the overhead of sharing m5 artifacts with partner teams and reduce the startup cost associated with fine-tuning M5 models for new downstream tasks.

Kingpin #283716:
Description: Vend access to M5 query representations through QUZen to one partner team. These representations should be made accessible at a rate of TK representations/s with a minimum retrieval latency of TK ms.
How is this Measured: Access is measured in terms of representations provided per second and minimum retrieval latency.
Why is this Important: The M5 project's ability to demonstrate value to partner teams depends on its ability to provide access to M5 representations. Without this, partner teams might instead consider developing their own models. Providing low-latency low-effort access to high-quality M5 representations lowers the bar to entry for prospective partner teams and improves our ability to drive impact in existing partnerships.

Kingpin #283674:
Description: Vend access to ASIN representations generated by M5 models through the Catalog Team's Universal Image Feature Catalog (UIFC) to TK partner teams. These representations will be based on multi-modal data consisting of images, text, and structured fields, derived from ASINs in TK languages from TK locales using data which is refreshed at a 15 minute cadence, and provided at a rate of TK representations/s with a minimum retrieval latency of TK ms. Additionally, provide an interface for partner teams to provide fine-tuning data and to select from a small set of pre-built fine-tuning strategies for customizing these representations to downstream tasks.
How is this Measured: Access is measured in terms of representations provided per second and minimum retrieval latency. Fine-tuning strategies include, for example, "number of ASINs presented as input", "loss function", "number of layers added on top of original m5 model", or "embedding type".
Why is this Important: The M5 project's ability to demonstrate value to partner teams depends on its ability to provide access to M5 representations. Without this, partner teams might instead consider developing their own models. Providing low-latency low-effort access to high-quality M5 representations lowers the bar to entry for prospective partner teams and improves our ability to drive impact in existing partnerships.

Kingpin #273867:
Description: Launch an ML model that incorporates M5 representations to map ASINs to brand entities in US, UK, DE, FR, IN marketplaces by 11/30.
How is this Measured: Demonstrate recall increase at 95% precision or reduction in model development and maintenance costs by incorporating M5 representations into a production system which relies on brand classification models in X marketplaces by 11/30.
Why is this Important: Brand is a key attribute used by systems across Amazon for use-cases such as ASIN and offer creation, merchandising, and selection gap monitoring. Mapping catalog items into their respective brand entities is a challenging problem due to inconsistencies in the item's brand attribute. The catalog team observe brands with multiple aliases, sellers mutating brand strings, sellers providing their shop names as the brands, brands with the same name, brand names that are common words, etc. Improving the ability to perform this task automatically both reduces the cost of maintaining this valuable data source and improves the quality of downstream tasks.

Kingpin #270645:
Description: Launch an ML model that incorporates M5 representations to identify duplicate ASINs in US, UK, DE, FR, IN marketplaces by 11/30.
How is this Measured: Demonstrate recall increase at 90% precision or reduction in model development and maintenance costs by incorporating M5 representations into a production system which relies on customer perceived duplicate models in X locales by 11/30.
Why is this Important: The Catalog Team's Item Authority project uses duplicate detection to identify sellers who list the same product using different ASINs across two or more marketplaces. The ability to detect these bad actors is critical to Amazon's "list once, sell globally mission" and improves its ability to produce a consistent user experience.

Kingpin #270649:
Description: Improve search quality in UAE, and one of TR and MX using semantic search powered by M5 representations.
How is this Measured: HERO scores for search results will improve as follows: in UAE from TK to TK, in TR from TK to TK, and in MX from TK to TK. This will be achieved by augmenting/generating the match set via online inference on a deep-learnt semantic matching model that is pre-trained in a locale and language agnostic way, and then fine-tuned to each locale. The semantic matching model will use M5 ASIN representations, again possibly fine-tuned for each locale, as input features.
Why is this Important: The Amazon Shopping experience primarily uses lexical representations of customer queries and the product catalog, supplemented with behavioral associations such as customer clicks, add to cart and purchase events. Behavioral data is absent for tail queries or in tail locales, which leads to low quality search results. For the Semantic Search team to achieve its goal of "Search Quality Parity Worldwide, from Day One" they will need to take advantage of richer representations of ASINs, queries, and customer behavior in order to provide a step function improvement in search quality. The M5 team is uniquely positioned to provide this data.

Kingpin #270652:
Description: Increase ASIN coverage of Top 25 attributes in Consumables, Top TK attributes in Softlines, Top TK attributes in Hardlines to at least 70% with at least 85% precision in DE, JP, FR, IT, ES using M5 representations.
How is this Measured: The PT list is manually identified from the 2021 pathfinder PT lists. Top attributes for each PT is identified

via RAI. ASIN coverage is SI-weighted. Precision of the fields is measured using human labels.
Why is this Important: To recognize exact matches for a spearfishing query such as ‘peets dark roast ground coffee 12oz’, we need QU to recognize the attributes in the query and DU to extract the attributes from the ASINs. In Hardlines and Softlines, we will build these DU models to supplement the work from Catalog and PG.

Kingpin #270648:
Description: Launch an ML model that incorporates M5 representations to improve the CTR prediction task in TK locales using M5 representations.
How is this Measured: Click conversion is measured according to online metrics measured against a baseline to be chosen by the Ads team.
Why is this Important: The Ads team uses a machine learning model to predict ads with the highest probability of click conversion for a given set of a search results. The CTR prediction team has experimented with BERT models for representing ASIN text and shown significant offline improvements (2.4% AUC lift). However, they have been unable to deploy these models to production yet due to the computational cost and high latency of BERT model inference. The M5 team is uniquely positioned to (1) extend this observation that larger models trained on more data lead to better CTR prediction results, and (2) produce models which are capable meeting the Ads team’s TK ms inference SLA.