Increasing Accessibility of Language Models with Multi-stage Information Extraction

Conrad Czejdo and Sambit Bhattacharya
Dept. of Mathematics and Computer Science, Fayetteville State University, Fayetteville, United States
Email: cczejdo1@broncos.uncfsu.edu, sbhattac@uncfsu.edu

***Abstract*—The capabilities of Language Models (LMs) have continued to increase in recent years, as have their computational requirements. Widely available APIs have also become available. These APIs present new challenges for ease of gradient based fine-tuning by users, resulting in the use models which may be larger than necessary and more expensive, therefore reducing accessibility. In this paper, we present a new methodology for increasing performance of single-shot LMs by chaining multiple smaller LMs. Additionally, as the derived representation is in plain-text it is readily human interpretable. We show that optimizing the context which leads to this derived representation results in improved performance and reduced cost.**

***Index Terms*—Deep Learning (DL), Natural Language Processing (NLP), Language Models (LM), one-shot learning, API**

I. Introduction

The goal of Language Models (LMs) is to predict the most likely text to be generated in a sequence given some form of context. The two most common forms of LMs are predicting future text from contextual, previous text (e.g. GPT [1], [2]) and filling in a blank from contextual, surrounding text (e.g. BERT [3], RoBERTa [4]). In recent years, LMs powered by neural network transformers have shown state-of-the-art performance across a multitude of language-reasoning tasks [5]. Although the full breadth of applications enabled by LMs is still an active area of research, many tools have become available to help disseminate the technology. Many platforms have developed their own solutions to help easily train and deploy large-scale LMs [6], [7]. However, direct under-the-hood access to large models is not provided across all services, which makes traditional forms of fine-tuning difficult. The decision to close models off is somewhat supported by the exponentially increasing burden of parameter size on available computational systems and the possibility of misuse. However, untrainable parameters can cause significant issues for finding a model which is both cost-effective and capable of solving a specific task. For example, the OpenAI API currently offers six AI models which cannot be readily fine-tuned. Each of the AI models is in different tiers of increasing cost and performance. The dilemma present is that inexpensive discrete options provided may not be sufficiently capable in modelling language, while the adequate models are too expensive. To this end, we investigate whether there are methods to utilize the functionality of multiple smaller LM models to reach the performance of a larger, more expensive model. Note that we use comparisons which are based on non-finetunable models available through the OpenAI API. With these APIs, it is possible to use either one shot or few shot examples. We focus our comparisons on one-shot learning for the following reasons: 1. it provides the least cost for the greatest number of tests (as the API charges based on input length), and 2. it is the most accessible version for users who do not have the ability of curating a dataset for a novel task, which are the primary target users for this research.

A. Multi-stage Language Models

Multi-stage models have been commonly developed for use in tasks such as image detection, where capturing variance in feature scales can be problematic [8], [9]. Furthermore, combining networks, such as Convolutional Neural Networks (CNNs) and logical reasoning solvers [10] has shown the ability of solving problems which CNNs alone cannot. In this work, we use multiple language models with their own assigned "tasks."

B. One-Shot Learning

Traditionally, AI models have required large training sets to learn from. This approach has two major difficulties for general audiences: 1. training very large AI models and 2. finding large datasets for specific tasks. Fine-tuning large, pre-trained models with gradient descent on smaller task-specific datasets has become a primary methodology of reducing the need of collecting extensive numbers of samples. The need for examples to fine-tune from, however, can still be large. Significant work has been done on developing methodologies for extremely low-sample training. The work has focused on few-shot (few examples), one-shot (one example), and even zero-shot (no example) training. Few-shot and one-shot learning were initially very challenging tasks with specific methodologies developed [11]-[13]. However, very large LMs, such as GPT-3, have shown to be particularly good at few-shot, one-shot, and in some cases, zero-shot training without using a specific training and inference paradigm [1]. If the context fed to the model

Manuscript received August 23, 2021; revised January 12, 2022.

© 2022 J. Adv. Inf. Technol.
doi: 10.12720/jait.13.2.181-185

contains an input-output pair for an example and ends in an input without an output, then GPT-3 will infer an output which logically follows from the first example. The accuracy of the model's output is based on its capability to understand the example and is restricted based on the capacity of the model (a larger model will be able to solve more difficult problems). The results from GPT-3 allow us to reasonably optimize our methodology using a single example for each "task" we assign to LMs, reducing the cost of testing each combination. Furthermore, only requiring a single example is more accessible than requiring users to curate larger datasets. In our work, we also utilize GPT-3's zero-shot capability to sample possible example derivations. This alleviates users from the need to write their own set of possible derivations.

C. Contributions

Our major contribution is an analysis of context optimizations for LMs including most appropriate examples for one shot learning and question syntax.

D. Organization

The rest of our paper is organized into the following sections. A Methodology (II) Section, where we introduce the dataset, a brief introduction to LMs, and the novel multi-stage model. A Results and Discussion (III) Section where we present our results from fine-tuning multiple parameters of the model and how they compare with baselines. Finally, we conclude with the implications of increased accessibility of our method, as well as future work.

II. Methodology

A. Data

To validate our method, we use the ReClor reading comprehension dataset [9]. ReClor suits our task due to: 1. short context, allowing for extensive testing on multiple parameters at once, 2. logic based questions which could benefit from a multi stage approach, and 3. continued difficulty that even very large, fine-tuned, language models have on the dataset.

ReClor is composed of questions from standardized graduate admissions examinations (e.g. GMAT and LSAT). Each sample has its own unique paragraph of context, a question, and four short-sentence answer choices (A, B, C, and D). We derive a simplified version of ReClor (ReClor_Simple) with balanced true/false answer choices taken from the original ReClor short-sentence answer choices. A Reclor_Simple question combines the original question with one of the short-sentence answer choices (A, B, C, or D), and if the answer choice is correct then the answer is true, otherwise it is false. An example of one sample question is shown in Fig. 1.

The total number of training examples available in Reclor Simple is 9256, and the total number of validation samples is 1000. Both sets of data are split fifty-fifty between positive (true) and negative (false) examples. To explore the potential of tuning multiple parameters, we further reduce the number of examples used for single shot testing to 25 random samples and 50 samples for validation.

```
In a business whose owners and employees all
belong to one family, the employees can be paid
exceptionally low wages. Hence, general operating
expenses are much lower than they would be for
other business ventures, making profits higher. So
a family business is a family' s surest road to
financial prosperity.
Is the following statement True or False?
"ignores the fact that in a family business,
paying family members low wages may itself reduce
the family's prosperity"
Answer: True
```

Figure 1. Best viewed in color. A sample from ReClor_Simple. In purple is the context, in green is the answer choice from ReClor and in red is the expected output (True or False).

B. Language Models

The problem of LMs is best formulated as modeling an unsupervised estimation of a distribution from a set of variable length sequences. Most language models use a Transformer architecture with multiple layers of attention [1], [5]. Attention is learned by learning a set of values (V) which construct a normalized output based on the dot product between a set of learned keys (K) and the query (Q). The most important feature of the Transformer architecture is the O(1) path length between input symbols. This is significantly different from RNNs which have O(input length) path length between input symbols. This advantage ensures the entire context is taken into account for each calculation, as well as increase prediction speed and training.

We use the GPT-3 class of language models (from smallest to largest: ada, babbage, curie, davinci) as well as available zero-shot fine-tuned versions (curie-instruct, davinci-instruct) [1], [7]. We set the hyperparameters of all models to a constant top P of 1.0 and temperature of 0.0, ensuring constant output across multiple API calls if the query is the same. This setup allowed for caching, thereby increasing sampling and reducing cost.

C. Proposed Multistage Language Model ("Staged")

We define an example (input, derivations, and output) to be the context of the inference LM which is optimized for the task. This context is optimized using the results of testing on multiple sample inputs from the dataset. Using the example input, derivations, and output we can generate derivations and an output for each sample in the dataset (Fig. 2).

Our hypothesis is that a smaller LM which outputs the expected response (True/False) as well as derived information about a sample can perform more accurately than a larger LM which only outputs the response (True/False). One of the restrictions of smaller GPT-3 LMs (ada and babbage) is the poor performance on zero shot tasks. Therefore, we utilize larger, fine-tuned LMs (curie-instruct or davinci-instruct) to sample possible example derivations. We also optimize the questions wording by altering syntax of what is extracted (e.g. "evidence," "details") and in what form (e.g. "bulleted,"

© 2022 J. Adv. Inf. Technol.

"list") Since sampling derivations from the large LM is only done while optimizing the multi-stage methodology on the dataset, and not during inference (Stages 2 and 3), we expect this stage will be a small fraction of production costs. Sampling and optimizing derivations could also be done through crowd-sourced methods but this would take far longer and be more tedious than collecting derivations through LM based generation. The derivation stage, called stage 1 (Fig. 2), is necessary since the smaller GPT-3 LM we use for inference do very poorly without at least one example. Therefore, we present an example derivation from a large LM for the smaller LM to use as context for its sample derivation (Stage 2).

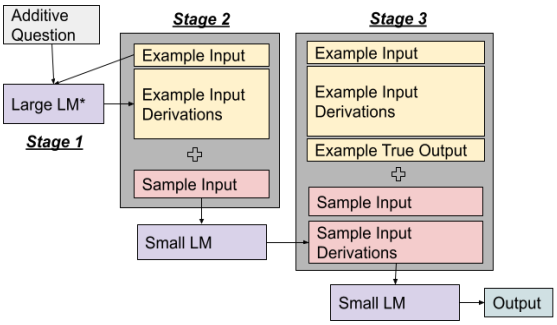

Figure 2. Best viewed in color. Stages of our multi-stage LM method. The color coding is as follows: Purple - LM, Red - input context and derivations, Yellow - denotes example context and derivations. The stages are as follows: Stage 1 - An optimized additive question is asked of a Large LM (*) which has been fine tuned for zero shot QA. Stage 2 - Example input and derivations from large LM are used as a single shot example for derivations from a sample. Stage 3 - The input and derivations and output from the example are used as a single shot example for the small LM to generate an output.

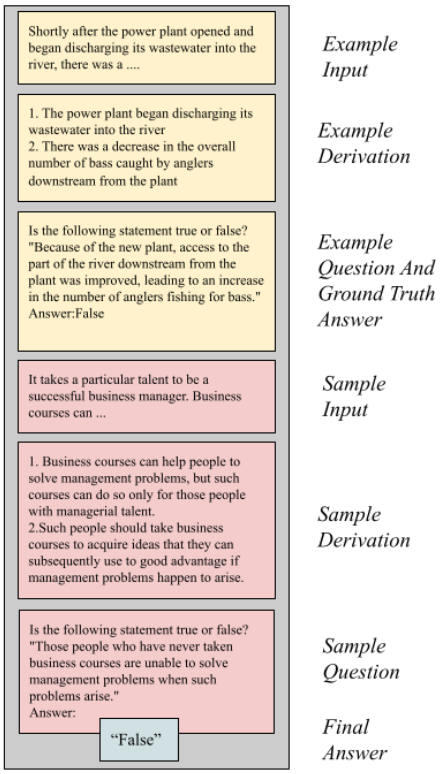

Figure 3. Best viewed in color. Example context and output of final stage (Stage 3 from Fig. 2).

In Stage 2, we define a single-shot example for a smaller LM using the example input and derivation from the large LM. In the final stage (Stage 3), the small LM uses a large context which includes: 1. the same example input and example derivation from Stage 2 with the example answer (ground truth known) appended, and 2. the sample input and sample derivation (from the LM output of Stage 2). This context makes the LM output an answer to the sample (True or False). In Fig. 3 we show an example of the context and output from the final stage.

We analyze three potential tunable parameters. The first tuning parameter is that of model size, as well as combinations of large (used in Stage 1) and small (used in Stage 2 and 3) sizes. The second tuning parameter is the syntax of the object of the question (e.g. "evidence," "details") for sampling potential example derivations (Stage 1).

The third tuning parameter is the list type (e.g. "bulleted," "list") of the question for sampling potential example derivations (also Stage 1).

III. Results and Discussion

A. LM Selection

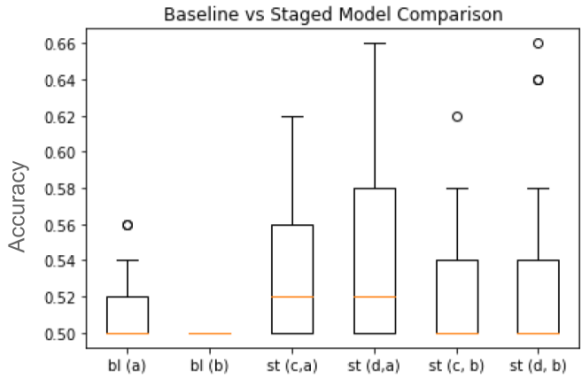

Figure 4. Box-and-whiskers plot comparison of simple baseline (bl) versus staged (st) models using 25 random single-shot examples. The same 25 single-shot examples were used for all tests. 50 validation samples were used to test each single-shot example. In parentheses are the models used: a stands for ada, b for babbage, c for curie, and d for davinci. The first model of two in the staged models stands for the large, zero-shot fine-tuned LM which is used for generating the first example (Fig. 2). The values of each average and max are as follows: model - (mean+/-s.d.), bl(a) - (0.51+/-0.02), bl(b) - (0.50+/-0.00), st(c,a) - (0.53+/-0.04), st(d,a) - (0.54+/0.04), st(c,b) - (0.53+/-0.03), st(d,b) - (0.53+/-0.05).

We first test how our staged model performs against a baseline model with no derived information (Fig. 4). Notable is that the baseline babbage model failed to produce results beyond random chance for any of the possible examples. This is interesting as it is a larger and more costly version of ada. Overall, ada showed a larger variance for the accuracy of each single shot example, which is a beneficial attribute of finding the single-shot example which leads to the highest accuracy during inference. Furthermore, extracting example information (Stage 1) with davinci showed slightly more accurate examples over extracting the same example information with curie, a slightly smaller model. Although our tests

© 2022 J. Adv. Inf. Technol.

show that there is no significant difference between the medians among all single-shot examples, in the case where the best example can be chosen there were much clearer differences. For example, our best staged example had an accuracy of 0.66 while our best baseline model had an accuracy of 0.56. Overall it is clear that deriving additional example information is important for improving the distribution of examples to pick from for models. Even babbage, which was picking randomly when unstaged, showed some examples with improved performance with a staged model.

B. *Effects of Interchanging Extracted Object Noun in Question*

Our next test was on optimizing the question provided to the zero-shot large LM initially (Fig. 4) in stage 1 of our model. Our results (Fig. 5) show that although there are some differences in the examples, the validation accuracy is largely resilient to changes in a single noun phrase.

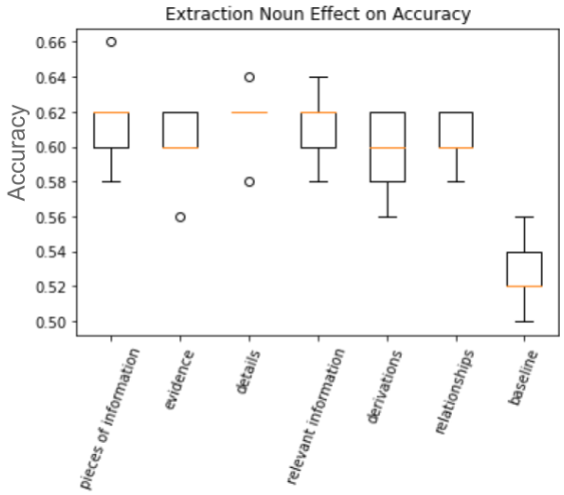

Figure 5. Box-and-whiskers comparison of what is asked to be extracted in the additive question. E.g. "details" stands for the additive zero-shot question "Extract a numbered list of details." We use the top model from Fig. 4 (davinci derivation and ada synthesis) and the top 5 examples ranked by accuracy on the validation set. We include the baseline of the top 5 ada examples as a comparison.

C. *Effects of Interchanging List Noun in Question*

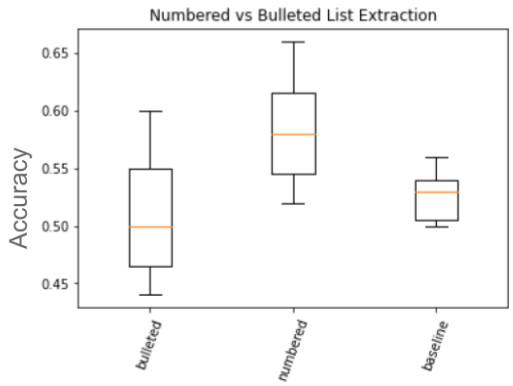

Figure 6. Box-and-whiskers comparison of bulleted (using '-') versus numbered list (using '1.', '2.', etc.) during example derivation (stage 2, Fig. 2).

We test using different representations for the model (Fig. 6), specifically in using numbered vs bulleted lists. We find that although the bulleted list manages to keep a larger distribution, it produces an overall negative shift in the accuracy of the samples. This indicates that although a relatively minor change, the list representation can have significant effects on the LM's internal ability to apply example logic to samples.

D. *Example Extractions*

We explore what types of derivations our models are capable of. Fig. 7 shows an example of a derivation with two notable features. The first feature is reducing the introduction in (list #1). The second is re-stating (list #2) and then adding a logical induction (list #3).

```
An experiment was done in which human subjects
recognize a pattern within a matrix of abstract
designs and then select another design that completes
that pattern. The results of the experiment were
surprising. The lowest expenditure of energy in
neurons in the brain was found in those subjects who
performed most successfully in the experiments.

1. Human subjects recognize a pattern within a matrix
of abstract designs and then select another design
that completes that pattern.

2. The lowest expenditure of energy in neurons in the
brain was found in those subjects who performed most
successfully in the experiments.

3. The highest expenditure of energy in neurons in
the brain was found in those subjects who performed
least successfully in the experiments.
```

Figure 7. Example extraction from the best performing model staged with davinci and ada.

We also see that many poor derivations exist, such as Fig. 8 from one of our worse performing bulleted, rather than numbered models. The derivation shows simple repetition not just in the first two bullets, but also a repeat across the last two bullets.

```
Britain is now rabies free. Nevertheless, Britain' s
strict quarantine of imported domesticated animals,
designed to prevent widespread outbreaks of rabies
there, cannot succeed indefinitely in preventing such
outbreaks. Bats, which are very susceptible to
rabies, fly into Britain from continental Europe.
Since wild bats cannot be quarantined, this policy
cannot control rabies spread by wild bats.

-Britain is now rabies free.
-The strict quarantine of imported domesticated
animals, designed to prevent widespread outbreaks of
rabies there, cannot succeed indefinitely in
preventing such outbreaks.
-Since wild bats cannot be quarantined, this policy
cannot control rabies spread by wild bats.
-Since wild bats cannot be quarantined, this policy
cannot control rabies spread by wild bats.ts.
```

Figure 8. Example extraction from a badly performing model staged with davinci and ada, but using bullets rather than numbers.

IV. Conclusion

As gradient-based methods of fine-tuning LMs become inaccessible due to exploding parameter sizes and debate surrounding ethical LM release policy, we demonstrate a new method for accessibly optimizing LMs for specific tasks. Our multi-stage method shows the utility of plain-text representations in increasing the accuracy of LMs. During our tests on our zero-shot stage (Stage 1), we find

© 2022 J. Adv. Inf. Technol.

that there are some types of question syntax for which LMs are resilient (Fig. 5) and also question syntax for which LMs can be significantly affected (Fig. 6).

This shows that even though LMs have been trained on an enormous corpus of text, that they can be biased in their ability to process input. This also shows why it is important to tune question syntax for each task. We show that a multi-stage LM variant provides the necessary variance to optimize single-shot learning through context rather than fine-tuning. This work increases LM accessibility by providing a method for tuning the output of smaller, more cost-effective, LMs by optimizing context.

For future work, we plan to extend our method by finding novel ways of optimizing question syntax to overcome the implicit biases found in LMs. A potential method is to leverage previous research in logical reasoning [14] and sentiment analysis [15]-[17] to score questions for their ability to yield better example derivations. A question scoring model would reduce the cost, both financially and time-wise, of testing large parameter models across many outputs. Another potential area of future work is to find "tasks" which small LMs are capable of performing when given single or few-shot results from zero-shot large LMs. In this work, the multi-stage method used tasks which were hand-defined. If such tasks could be defined by LMs, there could be more flexibility in the method and potentially higher accuracies after appropriate optimization.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

C. Czejdo conducted the research and wrote the paper; S. Bhattacharya analyzed the data, directed the research, and edited the paper; all authors approved the final version.

Acknowledgment

This article is based upon work supported by the National Science Foundation under Grant No. 1818694.

References

[1] A. Radford, *et al.*, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.
[2] T. Brown, *et al.*, "Language models are few-shot learners," arXiv:2005.14165, 2020.
[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "{BERT}: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171-4186.
[4] Y. Liu, *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.
[5] A. Vaswani, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6000-6010, 2017.
[6] H. Poor, *An Introduction to Signal Detection and Estimation*, New York: Springer-Verlag, 1985, ch. 4.
[7] HuggingFace. [Online]. Available: https://huggingface.co/
[8] OpenAI API Beta. [Online]. Available: https://beta.openai.com/
[9] H. Cheng, F. Liang, M. Li, B. Cheng, F. Yan, H. Li, V. Chandra, and Y. Chen, "ScaleNAS: One-Shot learning of scale-aware representations for visual recognition," arXiv:2011.14584, 2020.
[10] Z. Yan, *et al.*, "Bodypart recognition using multi-stage deep learning," in *Proc. International Conference on Information Processing in Medical Imaging*, 2015.
[11] P. Wang, P. Donti, B. B. Wilder, and Z. Kolter, "SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver," in *Proc. 36th International Conference on Machine Learning*, 2019.
[12] Y. Li, *et al.*, "DEEPre: Sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, pp. 760-769. 2017.
[13] F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594-611, April 2006.
[14] W. Yu, Z. Jiang, Y. Dong, and J. Feng, "ReClor: A reading comprehension dataset requiring logical reasoning," in *Proc. ICLR*, 2019.
[15] M. Darwich, *et al.*, "Quantifying the natural sentiment strength of polar term senses using semantic gloss information and degree adverbs," *Journal of Advances in Information Technology*, vol. 11, no. 3, pp. 109-118, August 2020.
[16] H. K. Darshan, A. R. Shankar, B. S. Harish, and H. M. K. Kumar, "Exploiting RLPI for sentiment analysis on movie reviews," *Journal of Advances in Information Technology*, vol. 10, no. 1, pp. 14-19, February 2019.
[17] X. Zhao and Y. Ohsawa, "Sentiment analysis on the online reviews based on hidden Markov model," *Journal of Advances in Information Technology*, vol. 9, no. 2, pp. 33-38, May 2018.

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Conrad Czejdo completed his undergraduate studies at the University of North Carolina at Chapel Hill in 2019, graduating with a degree of Bachelor of Science in Computer Science with Honors as well as a Bachelor of Arts in Chemistry. Currently, he is an MD candidate at the Western Michigan Homer Stryker University School of Medicine.
He has been awarded summer research internships at Lawrence Berkeley National Laboratory in 2016 and 2017. For the last several years he has been working as a research assistant for Dr. Sambit Bhattacharya at Fayetteville State University, Fayetteville, North Carolina. His research interests are in Data Analytics, Medical Sciences, and Natural Language Processing.

Sambit Bhattacharya is a Computer Scientist with more than 15 years of experience in teaching and research. He received his PhD in Computer Science and Engineering from the State University of New York at Buffalo, USA in 2005. He is a Professor in Computer Science at Fayetteville State University, North Carolina, USA.
He directs the Intelligent Systems Lab at Fayetteville State University. He has more than 50 peer reviewed publications and has delivered 40+ oral presentations, including keynote lectures at conferences. He leads projects funded by national funding agencies and has visited research labs of the US Department of Defense as a faculty fellow.
Dr. Bhattacharya is a Senior Member of the IEEE.