# B. TECH. PROJECT REPORT

On

# Improving the Efficiency of Large Scale Optimal Control Problems

BY

Ashutosh Bang and Kumar Abhinav



## DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING INDIAN INSTITUTE OF TECHNOLOGY INDORE December 2019

# **Improving the Efficiency of Large Scale Optimal Control Problems**

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree

of

#### **BACHELOR OF TECHNOLOGY**

in

### **COMPUTER SCIENCE AND ENGINEERING**

Submitted by:

Ashutosh Bang and Kumar Abhinav

Guided by:

Dr. Kapil Ahuja, Associate Professor, IIT Indore, and Dr. Marc Steinbach, Hannover University, Germany



INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2019

### **Declaration of Authorship**

We hereby declare that the project entitled "Improving the Efficiency of Large Scale Optimal Control Problems" submitted in partial fulfillment for the award of the degree of Bachelor of Technology completed under the supervision of Dr. Kapil Ahuja, Associate Professor, Computer Science and Engineering, IIT Indore and Dr. Marc Steinbach, Leibniz University Hannover is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Signed:

Ashutosh Bang Kumar Abhinav

### Certificate

This is to certify that the B.Tech Project entitled, "Improving the Efficiency of Large Scale Optimal Control Problems" and submitted by Ashutosh Bang and Kumar Abhinav in partial fulfillment of the requirements of B.Tech Project embodies the work done by them under my supervision.

Supervisor

Dr. KAPIL AHUJA

Associate Professor,

Indian Institute of Technology Indore

Date:

### Acknowledgments

It is our privilege to express our gratitude to several persons who helped us directly or indirectly to conduct this research project work. We express our heart full indebtedness to our BTP guide Dr. Kapil Ahuja and Dr. Marc Steinbach for his sincere guidance and inspiration in completing this Project.

We are extremely thankful to Mr. Aditya Anand Shastri for his coordination and cooperation and for his kind guidance and encouragement.

We also thank our friends who have more or less contributed to the making of this project.

This study has indeed helped us to explore more knowledgeable avenues related to this topic and we are sure it will help us in the future.

INDIAN INSTITUTE OF TECHNOLOGY INDORE

# Abstract

Department of Computer Science and Engineering

Bachelor of Technology

#### **Improving the Efficiency of Large Scale Optimal Control Problems**

Using SQP iteration, direct boundary value problem methods are significantly successful to get the solution of optimal control problems which are nonlinear. An application of it is Passive Solar Building in Germany. ODE (Ordinary Differential Equations) are used for solar cycle rooftop in Germany for maintaining water temperature in buildings.

The solution involves Conversion of ODEs to NLP (Non - Linear Programming) and then approximating the corresponding Non Linear Programming to Quadratic Programming problem using a special method called collocation. After that, the Quadratic Program formulation is solved using some specialized KKT routines. At present, MA27 is used as the solver for the respective specific Non Linear Programming problem.

Through this project, we aim at improvising the runtime of the algorithm by using Linear algebra techniques and a specialised KKT solver that works with dense matrices. We considered the significant requirement for ensuring the efficiency of the implementation as better implementation practices can lead to great time reductions.

# Contents

D	eclara	tion of Authorship	i	
С	ertific	ate	iii	
A	cknow	ledgements	v	
Abstract         Table of Contents         I Improving the Efficiency of Large Scale Optimal Control Problems         1 Introduction         1.1 Problem Statement         1.2 NLP Formulation         1.3 Collocation	vii			
T	able o	f Contents	viii	
Ι	Imj Pr	proving the Efficiency of Large Scale Optimal Control oblems	1	
1	Intr	oduction	1	
	1.1	Problem Statement	1	
	1.2	NLP Formulation	3	
	1.3	Collocation	3	
2	Solv	ing the NLP	4	
	2.1	Solution using MA27	4	
	2.2	Why overriding with MSKKT	5	
	2.3	Issues in overriding	5	
	2.4	Solution for Overriding	6	
	2.5	Achieving the solution	6	
	2.6	Results	7	
3	Effe	ctive Implementation	9	
	3.1	Challenges	9	
	3.2	Ensuring Implementation Efficiency	9	

4 Equations

.

4.1	Legend	10
4.2	Reformulating the Quadratic Program	11
4.3	Condensation Phase	12
4.4	Expansion Phase	14
4.5	Calculations	15
4.6	Analysis	16

5	Hun	nan Chatbot Interaction	17
	5.1	Introduction	17
	5.2	Analysis	19
	5.3	Part I: Machine Learning	20
	5.4	Part II: Platform	29
	5.5 Results		31
II	Futu	ıre Work	32

Π	References
	<b>I</b> UUUUUUU

33

# **List of Tables**

- $2.1 \ \ {\rm Copy, factorizing, solving and total time}$
- 2.2 Time complexity analysis

7 8

# Chapter 1

# Introduction

The history of optimal control problems dates back to the end of the seventeenth century, having its origin in the calculus of variations. For instance, the famous brachistochrone problem which was proposed by the Swiss mathematician Johann Bernoulli in 1696, maybe formulated as an optimal control problem. A specific theory of optimal control, however, has only been developed in the fifties and sixties of this century when real-life problems had to be solved in engineering and science. These textbooks and other early works study the questions of existence and uniqueness, controllability and attainability, and apply variational principles to derive necessary and sufficient optimality conditions for many problem classes. Extremal trajectories of a control problem are thus characterized as solutions of an associated boundary value problem. Numerical methods that rely on the solution of such boundary value problems are called indirect methods.

One real-life example of Large-Scale Optimal Control Problems is Passive solar building. Solar thermal energy is one good option for a sustainable way to provide heating or cooling for buildings or low-temperature processes.

## **1.1 Problem Statement**

ODE(Ordinary Differential Equations) are used for solar cycle rooftop in Germany for maintaining water temperature in buildings.

It involves

- Conversion of ODEs to NLP (Non-Linear Programming)
- Approximating NLP to QP using collocation.
- Solving the QP formulation using specialized KKT routines.



### **Passive Solar Building**

ImageRef:https://www.german-energy-solutions.de/GES/Redaktion/EN/Images/Articles/solar-h eating-energy-water-heating.jpg?\_\_blob=normal&v=3&size=834w

## **1.2 NLP Formulation:**

The problem statement can be formed in a Non-Linear Programming (NLP) format. There are solvers available to solve the corresponding NLP in an optimal manner like MA27, which is the "State of the Art" solver for the specific problem.

## **1.3 Collocation:**

Collocation, the second major BVP approach, differs from multiple shooting by the choice of local discretization schemes. On each subinterval (j; j+1), a polynomial  $p_j$  of a certain prescribed degree 'l' is taken as the local representation of the trajectory. This polynomial must satisfy the differential equation in each node of the local grid, which consists of 'l' collocation points.

# Chapter 2 Solving The NLP

## 2.1 Solution using MA27



MA27 is the solver for the specific NLP problem in which the expected input is in the triplet form. It is the state of the art solution for the corresponding problem.

- Compatible with sparse matrices.
- Existing solver for the corresponding NLP.

# 2.2 Why overriding with MSKKT(Multi-stage KKT)?

We are basically developing a model to create a wrapper for the process of overriding the state of the art MA27 solver with the MSKKT solver. The reasons supporting the corresponding operation are as follows:

- MSKKT is a fast specialized KKT solver
- Expected to significantly speed up the solving process.



## 2.3 Issues in overriding

While accomplishing the overriding of MA27 solver with MSKKT, there was a list of issues that were required to handle, to ensure the optimality of the corresponding operation. The issues are as follows:

- MSKKT works with dense matrices only.
- Due to collocation, matrices involved in NLP are highly sparse.
- We are wasting space as well as computation bypassing this directly over to MSKKT.
  - MSKKT is still only ~2s slower than MA27 even after this handicap.

# 2.4 Solution for Overriding

It was required to solve the underlying issues for ensuring the optimality of the overriding process of MSKKT. So, the strategies that we followed in order to solve the aforementioned issues are as follows:

- Making sparse matrices dense by eliminating collocation related matrices.
- Applying MSKKT on dense matrices for optimal results.
- Expanding the results by calculating values of collocation variables and corresponding multipliers



## 2.5 Achieving the Solution

The steps that we followed to resolve the issues that are encountered during the overriding of the MA27 solver by the MSKKT solver are as follows:

- Condensing of collocation conditions and variables efficiently.
- By developing a model for MSKKT wrapper with MA27 interface.
- Implemented in C.

## 2.6 Results

The results that we got after the overriding of the state of the art MA27 solver by the MSKKT solver in terms of the factorization time, solving time, copying time and the total time for these operations are formatted in a tabular form. There was a reduction in the number of variables involved during the solving of equations drastically.

- The number of total variables reduced from:
  - o 1627020 -> 122616
  - $\circ$  Reduced by ~13 times.
- RHS(Right Hand Side) variables reduced from:
  - o 35370 -> 9432
  - $\circ$  Reduced by ~4 times.

For an explanation of how the above reduction was achieved, refer Section 4.5

Algorithm	Values	Time Complexity	Explanation
spd_mul	sparse matrix dimn: n*m no of non-zeroes: nz dense matrix dimn: m*k	$O(nz^*m + n^*k)$	Sparse matrix * dense matrix
dsp_mul	dense matrix dimn: n*m sparse matrix dimn: m*k no of non-zeroes: nz	$O(nz^*m + n^*k)$	Dense matrix * sparse matrix
mat_add	Matrix dimension: n*m	O(n*m)	Add two dense matrices
mat_mul	Matrix dimension:n*m, m*k	O(n*m*k)	Multiply two dense matrices
transpose	Matrix dimension: n*m	O(n*m)	Transpose a dense matrix
densify	Matrix dimension: n*m	O(n*m)	Convert sparse matrix to a column major dense matrix
setJ	Resultant matrix dimension: nx*(nx+nu)	O(nx*(nx+nu))	set control matrix from calculated values (for use in MSKKT)
setH	Resultant matrix dimension: (nx+nu)*(nx+nu)	O((nx+nu)*(nx+nu))	set state matrix from calculated values (for use in MSKKT)

Table 2.1: Time complexity analysis

Solver	No of iterations	Min. objective function	Timings (s)
MA27	152	2.91037748446228 74e+05	15.695
MSKKT (without condensation)	154	2.91037748446227 87e+05	copy 0.86 factorize 13.34 solve 2.95 total 17.15 Per iteration:- copy 0.006 factorize 0.087 solve 0.019 total 0.111
MSKKT (after condensation)	4	4.13635391624223 38e+05	copy 1.31 factorize 0.02 solve 0.13 total 1.46 Per iteration:- copy 0.3275 factorize 0.005 solve 0.0325

 Table 2.2: Runtime Performance analysis

Note: Copy time increase because of calculation of the condensed matrix. In the initial phase, all 3 matrices (H, C and G) need to be condensed. However, in the subsequent stages, only the Hessian matrix requires recondensing. Hence, the overhead of copy would reduce with increase in the number of iterations.

# Chapter 3

# **Effective Implementation**

There were a lot of challenges involved during the process of improving the efficiency of Large Scale Optimal Control Problems. There was a strict requirement of ensuring the implementation efficiency for optimal results. We need to execute a list of operations on the matrices while maintaining the sparsity of the matrices involved in the equation. The details of the corresponding topics is discussed in the following section.

# 3.1 Challenges:

The hurdles in the process of improving the efficiency of large scale optimal control problems are as follows:

- Maintaining sparsity of the matrices until the number of zeros was negligible.
- Applying complex operations on sparse matrices while maintaining the existing format:
  - LU factorization
  - Inverse operations
- Implementing routines that efficiently handle operations between sparse and dense matrices.
- Handling a lot of pointers ensuring there is no memory leak

# **3.2 Ensuring Implementation efficiency**

There was a significant requirement for ensuring the efficiency of the implementation as better implementation practices can lead to great time reductions. The steps that we followed to ensure the same is as follows:

- Reducing the number of memory allocation (malloc, calloc) calls as much as possible, by creating buffers at initialization, and efficiently reusing the space to cut down on allocation costs and heap fragmentation.
- Maximizing Cache efficiency of the solution, for example by leveraging cache locality in matrix multiplications by rearranging the order.
- Multipass calculations to evaluate and set values in MSKKT buffer, which avoids the overhead of going back and forth to set them.

# Chapter 4 Equations

The reformulation takes part in two phases. First, we condense the matrices and rhs vectors, eliminating all the terms corresponding to the collocation variables. Then, these matrices and vectors are sent to be evaluated by MSKKT, which then returns the corresponding values of state and control variables and multipliers (specifically, x, u and  $\lambda$ ). These values are then utilized to 'expand' the results by calculating the collocation variables and multipliers (z and  $\mu$ ). The corresponding formulation to achieve the objective are as follows:-

## 4.1 Legend

- 1. x = Initial State Variables
- 2. u = Control Variables
- 3. z = Collocation Variables
- 4. H = Hessian Matrix
- 5. G = Control Equality Constraints
- 6. C = Collocation Equality Constraints
- 7. h = RHS vector of control equality constraints
- 8. c = RHS vector of collocation equality constraints

## **4.2 Reformulating the Quadratic Program** 1.

$$\begin{split} \min_{\mathbf{x},\mathbf{z},\mathbf{u}} & \sum_{j=0}^{m} \left[ \frac{1}{2} \begin{pmatrix} \mathbf{x}_{j} \\ \mathbf{z}_{j} \\ \mathbf{u}_{j} \end{pmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{H}_{jx}^{\mathsf{xx}} & \mathbf{H}_{jz}^{\mathsf{xx}} & \mathbf{H}_{ju}^{\mathsf{xx}} \\ \mathbf{H}_{jx}^{\mathsf{xx}} & \mathbf{H}_{ju}^{\mathsf{xx}} & \mathbf{H}_{ju}^{\mathsf{xx}} \\ \mathbf{H}_{ju}^{\mathsf{xx}} & \mathbf{H}_{ju}^{\mathsf{xx}} & \mathbf{H}_{ju}^{\mathsf{xx}} \\ \mathbf{x}_{j} \\ \mathbf{x$$

Formulation reference: www.ifam.uni-hannover.de/fileadmin/IFAM/ordner/steinbach /publications/dipl\_diss/urz.ps

## 4.3 Condensation Phase



Mathematically it is a projection of several sparse matrices on the null space of another sparse matrix, where the resulting projected matrices are dense and are to be placed into a special KKT solver, which is called MSKKT.

Recall that the local discretization scheme in collocation is solved as part of the global NLP, and hence in each SQP iteration as part of the QP. Compared to multiple shooting on the same global grid, the QP is, therefore, larger, but it has a more specific structure.

More precisely, the blocks sitting at the corners of derived matrices are also present in multiple shooting, whereas all the central blocks, being associated with collocation variables and conditions, belong solely to the local discretization. In the following we will show how collocation variables and conditions are locally eliminated, leaving a condensed QP which has exactly the same structure as in multiple shooting.

$$\bar{\mathbf{f}}_{i}^{u} := \mathbf{f}_{i}^{u} - \mathbf{H}_{i}^{uz} \bar{\mathbf{c}}_{j} + (\bar{\mathbf{C}}_{j}^{u})^{\mathsf{T}} \bar{\mathbf{f}}_{i}^{z}$$

$$\begin{split} \bar{\mathbf{h}}_j &\coloneqq \mathbf{h}_j - \mathbf{G}_j^z \bar{\mathbf{c}}_j, \\ \bar{\mathbf{f}}_i^x &\coloneqq \mathbf{f}_i^x - \mathbf{H}_i^{xz} \bar{\mathbf{c}}_j + (\bar{\mathbf{C}}_j^x)^\mathsf{T} \bar{\mathbf{f}}_i^z, \end{split}$$

$$\begin{split} \bar{G}_{j}^{x} &\coloneqq G_{j}^{x} - G_{j}^{z}\bar{C}_{j}^{x}, \\ \bar{G}_{j}^{u} &\coloneqq G_{j}^{u} - G_{j}^{z}\bar{C}_{j}^{u}, \\ \bar{H}_{i}^{xx} &\coloneqq H_{i}^{xx} - H_{i}^{xz}\bar{C}_{j}^{x} - (\bar{C}_{j}^{x})^{\mathsf{T}}H_{i}^{zx} + (\bar{C}_{j}^{x})^{\mathsf{T}}H_{i}^{zz}\bar{C}_{j}^{x}, \\ \bar{H}_{i}^{ux} &\coloneqq H_{i}^{ux} - (\bar{C}_{j}^{u})^{\mathsf{T}}H_{i}^{zx} - H_{i}^{uz}\bar{C}_{j}^{x} + (\bar{C}_{j}^{u})^{\mathsf{T}}H_{i}^{zz}\bar{C}_{j}^{x}, \\ \bar{H}_{i}^{uu} &\coloneqq H_{i}^{uu} - H_{i}^{uz}\bar{C}_{j}^{u} - (\bar{C}_{j}^{u})^{\mathsf{T}}H_{i}^{zu} + (\bar{C}_{j}^{u})^{\mathsf{T}}H_{i}^{zz}\bar{C}_{j}^{u}. \end{split}$$

## **4.3 Expansion Phase**



 $C_j^z = L_j U_j$ 

 $z_{i} = U_{j}^{-1} L_{j}^{-1} [c_{j} - C_{j}^{x} x_{i} - C_{j}^{u} u_{i}] \Longrightarrow \bar{c}_{j} - \bar{C}_{j}^{x} x_{i} - \bar{C}_{j}^{u} u_{i}$  $-\mu_{j} = L_{j}^{-T} U_{j}^{-T} [f_{i}^{z} - H_{i}^{zx} x_{i} - H_{i}^{zz} z_{i} - H_{i}^{zu} u_{i} - (G_{j}^{z})^{T} (-\lambda_{j})]$ 

# 4.4 Calculations

The following equations explains how we arrived at the above formulation:-

Equations :

$$\begin{split} -\mu_{j} &= L_{j}^{-T} U_{j}^{-T} [f_{i}^{z} - H_{i}^{zx} x_{i} - H_{i}^{zz} z_{i} - H_{i}^{zu} u_{i} - (G_{j}^{z})^{T} (-\lambda_{j})] \\ &= L_{j}^{-T} U_{j}^{-T} [(f_{i}^{z} - H_{i}^{zz} \bar{c}_{j}) - (H_{i}^{zx} - H_{i}^{zz} \bar{C}_{j}^{x}) x_{i} - (H_{i}^{zu} - H_{i}^{zz} \bar{C}_{j}^{u}) u_{i} - (G_{j}^{z})^{T} (-\lambda_{j})] \\ &=: L_{j}^{-T} U_{j}^{-T} [\bar{f}_{i}^{z} - \bar{H}_{i}^{zx} x_{i} - \bar{H}_{i}^{zu} u_{i} - (G_{j}^{z})^{T} (-\lambda_{j})]. \end{split}$$

$$0 = x_i + G_j^z z_i - x_j - h_j$$
  
=  $(G_j^x - G_j^z \bar{C}_j^x) x_i + (G_j^u - G_j^z \bar{C}_j^u) u_i - x_j - (h_j - G_j^z \bar{c}_j)$   
=:  $\bar{G}_j^x x_i + \bar{G}_j^u u_i - x_j - \bar{h}_j$ ,

$$\begin{split} 0 &= \mathsf{H}_{i}^{xx} x_{i} + \mathsf{H}_{i}^{xz} z_{i} + \mathsf{H}_{i}^{xu} u_{i} - (-\lambda_{i}) + (-\lambda_{j}) + (C_{j}^{x})^{\mathsf{T}} (-\mu_{j}) - f_{i}^{x} \\ &= (\mathsf{H}_{i}^{xx} - \mathsf{H}_{i}^{xz} \bar{\mathsf{C}}_{j}^{x}) x_{i} + (\mathsf{H}_{i}^{xu} - \mathsf{H}_{i}^{xz} \bar{\mathsf{C}}_{j}^{u}) u_{i} - (-\lambda_{i}) + (-\lambda_{j}) + (C_{j}^{x})^{\mathsf{T}} (-\mu_{j}) - (f_{i}^{x} - \mathsf{H}_{i}^{xz} \bar{\mathsf{c}}_{j}) \\ &=: \tilde{\mathsf{H}}_{i}^{xx} x_{i} + \tilde{\mathsf{H}}_{i}^{xu} u_{i} - (-\lambda_{i}) + (-\lambda_{j}) + (\bar{\mathsf{C}}_{j}^{x})^{\mathsf{T}} [\bar{f}_{i}^{z} - \bar{\mathsf{H}}_{i}^{zx} x_{i} - \bar{\mathsf{H}}_{i}^{zu} u_{i} - (G_{j}^{z})^{\mathsf{T}} (-\lambda_{j})] - \tilde{f}_{i}^{x} \\ &=: \bar{\mathsf{H}}_{i}^{xx} x_{i} + \bar{\mathsf{H}}_{i}^{xu} u_{i} - (-\lambda_{i}) + (\bar{\mathsf{G}}_{j}^{x})^{\mathsf{T}} (-\lambda_{j}) - \bar{f}_{i}^{x}, \end{split}$$

$$\begin{split} 0 &= H_i^{ux} x_i + H_i^{uz} z_i + H_i^{uu} u_i + (C_j^u)^T (-\mu_j) - f_i^u \\ &= (H_i^{ux} - H_i^{uz} \bar{C}_j^x) x_i + (H_i^{uu} - H_i^{uz} \bar{C}_j^u) u_i + (C_j^u)^T (-\mu_j) - (f_i^u - H_i^{uz} \bar{c}_j) \\ &=: \tilde{H}_i^{ux} x_i + \tilde{H}_i^{uu} u_i + (\bar{C}_j^u)^T [\bar{f}_i^z - \bar{H}_i^{zx} x_i - \bar{H}_i^{zu} u_i - (G_j^z)^T (-\lambda_j)] - \tilde{f}_i^u \\ &=: \bar{H}_i^{ux} x_i + \bar{H}_i^{uu} u_i + (\bar{G}_j^u)^T (-\lambda_j) - \bar{f}_i^u. \end{split}$$

## 4.5 Analysis

Dimensions for the current problem:-

- 1. No of stages (m) = 393
- 2. Dimension of x = 11
- 3. Dimension of z = 33

Therefore, dimensions and no of entries are:-

Without Expansion	With Expansion
$H = 46 \times 46$	$\overline{\mathrm{H}} = 13 \mathrm{ x} 13$
G = 11 x 46	$\overline{G} = 11 \times 13$
$C = 33 \times 46$	
x = 11 x 1	x = 11 x 1
z = 33 x 1	
$\mathbf{u} = 2 \mathbf{x} 1$	$u = 2 \ge 1$
$\mu = 33 \times 1$	
$\lambda = 11 \text{ x } 1$	$\lambda = 11 \times 1$

Matrices: 1627020 Vectors: 35370 Matrices: 122616 Vectors: 9432

# Chapter 5 Human Chatbot Interaction

## **5.1 Introduction**

We are trying to analyze how humans converse with other chat-bots, to try and understand how we differentiate between humans and bots. Understanding this better would result in significantly enhancing the user experience, by improving response quality to mimic human conversations, leading to more engaging conversations.

Some examples of how this would benefit certain use-cases would be:

Duolingo: It is a platform that people use to learn the native language. To enhance the learning speed, conversing with someone well-versed in that language is highly beneficial. However, the non-native speakers are generally shy, since they are prone to making mistakes in the language they are learning. To solve this, Duolingo introduced chat-bots, which the users could talk to. This study could help the developers to make the model behave more like a native speaker, improving the user experience

Grammarly: Grammarly helps to grammatically correct sentences. The study could help by possibly improving the tonality of their suggestions.

IITI Website: A chat-bot would greatly enhance a user's experience of our institute's website, by making content more accessible. Having closer to human interaction would be a bonus!

Since we were interested in understanding the limitations better, we tried looking at related queries on the internet. Though there are certain well-known shortcomings, there has not yet been a formal study on the topic, which makes our project the first of its kind.



#### Why the best dataset for a suitable model?

- Gives great results on a small dataset.
- Dataset gives generic talks
- For the CSE website:
  - The manual dataset will be small
  - We wish to have a generic talk
- Suitable for Human Chatbot Interaction due to Generic Talks.

⊙ 2 minutes ago Hi	Mr. Human
<b>Mr. Robot</b> Hello there!	🕒 2 minutes ago
© 2 minutes ago How are you doing?	Mr. Human
<b>Mr. Robot</b> Never felt better	🕑 2 minutes ago
O 2 minutes ago What is the meaning of life?	Mr. Human
Ma Dahat	O a minute ago
I don't understand what you mean	e annote ago
I don't understand what you mean	
I don't understand what you mean	
I don't understand what you mean	
I don't understand what you mean	

## 5.2 Analysis:

We are developing a chat app to analyze the conversations between humans and the chat-bot. We first analyze the different categories of chatbots, picking up the most suitable for our use-case. We then proceed with using a suitable dataset, and applying several NLP pre-processing techniques. Multiple state-of-the-art Algorithms are then compared with each other, and the one having the Most Natural response was selected. We then integrate it with the chat app platform that will allow the chatbot to converse with other humans.

## 5.3 Part I: Machine Learning

### **Different Types of Chatbots:**

Refer to the image below



Image Ref: <u>https://miro.medium.com/max/704/1\*r8rR34sfjX4zdM0tXOsvaQ.png</u> Retrieval-Based models generate responses from a **fixed dataset**. This prevents it from being a truly 'open domain' solution since we cannot possibly store all the responses. However, it can be reasonably approximated by storing the majority of common discussions. A reasonable justification that we could provide is that humans don't really know a proper response to everything, hence we do not really expect someone else to be all-knowing either.

Generative models are used to generate responses *on the fly*. However, they are highly prone to making grammatical errors. We also do not yet have the necessary compute and machine learning models to solve the open domain version.

### [REF#1] Anatomy of a Chatbot



12

#### Image Ref: https://miro.medium.com/max/782/1\*4SzjHTccgX85iRrw589Y1g.png

#### **Brief Description of Natural Language Processing:**

Natural Language Processing (abbreviated as NLP) is the study of the interaction between computers and human languages. It is a fusion of Computer Science, AI, and Computational linguistics. Key uses include:

- Language translation applications such as Google Translate
- Word Processors such as Microsoft Word and Grammarly that employ NLP to check the grammatical accuracy of texts.
- Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as Google Assistant, Siri, Cortana, and Alexa.

#### **Preprocessing the Data:**

Converting raw data into trainable vectors undergoes the following series of operations

- Converting the entire text into **uppercase or lowercase**, so that the algorithm does not treat the same words in different cases as different
- Tokenization: Converting strings into a list of tokens i.e words
- Removing Noise, i.e everything that isn't in a standard number or letter.
- Removing **Stop words**.
- **Stemming**: Process of reducing inflected (or sometimes derived) words to their stem, base or root form generally a written word form.
  - For example, if we were to stem the following words: "Stems", "Stemming", "Stemmed", and "Stemtization", the result would be a single word "stem".
- Lemmatization: A slight variant of stemming. The major difference: Stemming can often create non-existent words, whereas lemmas are actual words. So, your root stem, meaning the word you end up with, is not something you can just look up in a dictionary, but you can look up a lemma. Examples of Lemmatization are that "run" is a base form for words like "running" or "ran" or that the word "better" and "good" are in the same lemma so they are considered the same.
- TF-IDF:
  - Problem: Highly frequent words start to dominate in the document (e.g. larger score), but may not contain as much "informational content".
  - Give more weight to longer documents than shorter documents.
  - Approach: Rescale the frequency of words by how often they appear in all documents.
  - So that the scores for frequent words like "the" that are also frequent across all documents are penalized. This approach to scoring is called Term Frequency-Inverse Document Frequency, or TF-IDF for short, where:
    - Term Frequency: is a scoring of the frequency of the word in the current document.
    - TF = (Number of times term t appears in a document)/(Number of terms in the document)
    - Inverse Document Frequency: is a scoring of how rare the word is across documents.
    - IDF =  $1 + \log(N/n)$ , where, N is the number of documents and n is the

number of documents a term t has appeared.

- The product **TF\*IDF** is then used to analyze the significance of a word in the document
- Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.

#### Why do we need them?

Consider the following similar sentences: *Have a good day* and *Have a great day*. They hardly have a different meaning. If we construct an exhaustive vocabulary (let's call it V), it would have  $V = \{Have, a, good, great, day\}$ .

Now, let us create an encoded vector for each of these words in V. Length of our encoded vector would be equal to the size of V (=5). We would have a vector of zeros except for the element at the index representing the corresponding word in the vocabulary. That particular element would be one. The encodings below would explain this better.

Have = [1,0,0,0,0]'; a=[0,1,0,0,0]'; good=[0,0,1,0,0]'; great=[0,0,0,1,0]'; day=[0,0,0,0,1]'(' represents transpose)

If we try to visualize these encodings, we can think of a 5-dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions). This means 'good' and 'great' are as different as 'day' and 'have', which is not

true. Our objective is to have words with similar context occupy close spatial positions. Mathematically, the cosine of the angle between such vectors should be close to 1, i.e. angle close to 0.



Image Ref: http://i0.wp.com/techinpink.com/wp-content/uploads/2017/07/cosine.png

Here comes the idea of generating *distributed representations*. Intuitively, we introduce some *dependence* of one word on the other words. The words in the context of this word would get a greater share of this *dependence*. In encoding representations, all the words are *independent* of each other, as mentioned earlier.

#### Survey of Datasets:

Major Requirement: Since we are aiming for natural human interaction, we require the dataset to be **conversational.** This requirement disqualifies some well-known datasets like Twitter Customer Support, Ubuntu Corpus, Wiki QA Corpus.

Following are some well known conversational datasets:-

- Reddit Dataset:
  - By far the biggest available dataset for language processing. However, training would take months to provide a reasonable response, and reducing the size of the dataset would destroy the diversity of responses. This leaves us with

experimenting on some pre-trained models, which led us to realize profanity in the dataset being a big issue.

- IRC Chat Logs :
  - The audience was mostly of a technical background, which would bias the model
- ConvAI2:
  - Consists of conversations between bots and humans, the model ended up learning the flaws of the bots involved in the conversation, when used alone.
- Cornell Movie Dialogue Corpus:
  - Consists of discussions between movie characters. The conversation is general but superficial in nature.

We currently tried using the ConvAI2 dataset, mixed with some external data to reduce the degree of error, and achieved mostly convincing responses from the model.

### **Survey of Models**

#### ChatterBot

- Most simple model
- Language independent design
- Generate responses based on collections of known conversations
- Selects the closest matching response by searching for the closest matching known statement that matches the input, it then returns the most likely response to that statement based on how frequently each response is issued by the people the bot communicates with.
- Simply uses word2vec
  - For retrieving the most appropriate response
  - Finding the most similar context

#### ChatBot RNN

- The underlying model is a character-based sequence predictor
- Uses optional beam search
- The highest probability word is selected as the output by the decoder. But this does not always yield the best results, because of the basic problem of greedy algorithms. Hence beam search is applied which suggests possible translations at each step. This is done by making a tree of top k-results.
- Uses relevance masking/MMI( maximum mutual information) to formulate its responses







- Provided the context and response, the model replies with the probability of the response being valid for the context
- Runtime is linear in the number of responses
- So, scalability is a big issue.

#### Seq2Seq



Image Ref: https://www.guru99.com/images/1/111318\_0848\_seq2seqSequ4.png

- Uses LSTM
- Encoder: It uses deep neural network layers and converts the input words to corresponding hidden vectors. Each vector represents the current word and the context of the word.
- **Decoder:** It is similar to the encoder. It takes as input the hidden vector generated by encoder, its own hidden states and current word to produce the next hidden vector and finally predict the next word.

Which model to choose?

The decision finally boiled down to using Seq2Seq (Generative) v/s Chatterbot (Rule-based), which we found to be the best of its respective categories. We observed that the Generative model was throwing grammatical errors and was not able to maintain the context unless it was trained with large amounts of data. Because of its simple design, **Chatterbot** performed well on the medium-sized dataset (**ConvAI2**), hence it is our **current mode**l.

## 5.4 Part II: Platform

### AIM:

We want to design a chat application that allows user-user and user-bot blind interaction. The app would require humans to interact with another person/bot, and identify whether there is a bot on the other end. We will analyze the accuracy of the claim, as well as ask users who identified the bot correctly about how they managed to identify it.

The Github Repo for our app could be found at https://github.com/Abhinav2812/chat-app

#### Tech Stack:

- ✤ Javascript
  - ➤ React & Redux (For the app's front-end)
  - Express JS (For the back-end, pairing the user as well as interfacing with the chatbot AI)
  - Socket.IO (For instantaneous messaging)
- Python
  - > Tensorflow & Keras: For the machine learning models
  - > NLTK: For pre-processing the data
  - ► Flask: For hosting the chatbot API

### **Responsive Design:**

Type your message here...

😵 🖼 🖸 💽 🚍 🖽 🦻

The app is compatible with devices of different form factors like smartphones, tablets, PCs, etc.

	⊙ 2 minutes ago Hi	Mr. Human		
	<b>Mr. Robot</b> Hello there!	② 2 minutes ago		
	<ul><li>☑ 2 minutes ago</li><li>How are you doing?</li></ul>	Mr. Human		
	<b>Mr. Robot</b> Never felt better	② 2 minutes ago		
	O 2 minutes ago What is the meaning of life?	Mr. Human		
	<b>Mr. Robot</b> I don't understand what you mean	C a minute ago		
	Type your message here	Send		
Iocalhost:3000 × +				-08
$\leftrightarrow$ $\rightarrow$ C (i) localhost:3000		x 0	0 II N 💀 🛛	s 🚳 💿 🚺 🚥 🛛 😨 🔅
⊙ a minute ago Hi				Mr. Human
Mr. Robot Hello there!				🕲 a minute ago
⊙ a few seconds ago How are you doing?				Mr. Human
<b>Mr. Robot</b> Never felt better				O a few seconds ago
O a few seconds ago What is the meaning of life?				Mr. Human
<b>Mr. Robot</b> I don't understand what you mean				O a few seconds ago

📕 📽 Wed Sep 25, 15:08:05 🎅 🕪 🕻 0% 🕶

# 5.5 Results:

Responses generated by the model were highly convincing. Some of the responses are as below:-

localhost:3000 × +	
D C Q localhost:3000	👰 사 🛞 :
) 16 minutes ago rhat do you do?	Mr. Human
Ir. Robot am a very experienced person	O 16 minutes ago
) 16 minutes ago 1 which field	Mr. Huma
Ir. Robot am a psych major what about you	O 16 minutes ag
) 16 minutes ago ndergraduate at IIT	Mr. Humai
Ir. Robot /hat about you	O 15 minutes ag
) 15 minutes ago achelor	Mr. Huma
Ir. Robot am your life	O 15 minutes ag
ir. Robot hat is Al?	O 15 minutes ag
) a minute ago ne of the best inventions of man	Mr. Huma
r. Robot Is interesting	O a minute ag
Type your message here	Sen
	🖌 🦉 🙀 Wed Sep 25, 20:25:52 • 🍷 🌒 🕃 👀
localhost:3000 × +	
D C □ 0 localhost:3000	👽 🚣 🛛 🛞 🗄

I don't understand what you mean	
Ø 3 minutes ago hi	Mr. Human
Mr. Robot How are you doing?	© 3 minutes ago
O 3 minutes ago I'm doing great!	Mr. Human
Mr. Robot Lagree.	© 3 minutes ago
O 3 minutes ago Nice	Mr. Human
Mr. Robot i am a huge gamer my mom is a very good person	© 3 minutes ago
O 2 minutes ago Nice to hear	Mr. Human
Mr. Robot Yes it is.	© 2 minutes ago
O 2 minutes apo What is the meaning of life?	Mr. Human
Mr. Robot There are other entities who are better versed on the topic.	O z minutes ago
Type your message here	Send
	₩ed Sep 25, 19:59:18 ● 🎅 🐠 😥 🕬

# **Future work:**

- LU factorization routine takes ~90% of the entire condensation/expansion time overhead. Improving this aspect will lead to further significant speed enhancements.
- Parallelization of the condensation/expansion routines, since the coupling is only between adjacent stages.
- Human Chatbot Interaction
  - Machine Learning
    - Analyze the conversations
    - We observed that restricting the length of responses improved the performance of the generative model
    - Hunting for more observations other than:
      - Losing context during a long conversation
      - Elapsed time for providing responses
  - Platform
    - To support the random assignment of users and bots
    - To make the web-app installable: Would provide a seamless cross-platform native experience
    - To allow for both one-to-one and group chats

# **References:**

- 1. https://lionbridge.ai/datasets/15-best-chatbot-datasets-for-machine-learning/
- 2. https://www.reddit.com/r/MachineLearning/comments/51x7px/p\_pretrained\_rnn\_chatbot/
- 3. https://github.com/Janinanu/UDC\_Chatbot/
- 4. https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c 8c8215ac6e
- 5. https://github.com/gunthercox/ChatterBot
- 6. https://arxiv.org/abs/1409.3215
- 7. https://arxiv.org/abs/1301.3781
- 8. https://people.sc.fsu.edu/~jburkardt/c\_src/csparse/csparse.html
- 9. https://github.com/ibayer/CSparse
- 10. https://www.ifam.uni-hannover.de/fileadmin/IFAM/ordner/steinbach/publications/dipl\_di ss/urz.ps
- 11. https://pdfs.semanticscholar.org/0cd4/c5db09cffa50b20db73f2c6d2485e2bb7d79.pdf