Pitch A: A worthwhile area of research would be investigating effective means to censor any undesirable responses an AI chatbot might produce. This would serve a similar purpose to the first avenue of research, but rather than keeping bad data out, this approach would center around keeping bad responses in. Training an AI on large data sets of logged conversations from the internet will inevitably lead to the chatbot training in some responses that are offensive/obscene. Filtering and moderating large data sets may prove to be cost-prohibitive, so it may instead be fruitful to make efforts to moderate the AI's responses. It may very well be easier to find a way to detect that a generated response is undesirable and prevent it from being displayed at all. There are a variety of techniques suitable for this end, black listing words that should never be used in any context, developing an algorithm to score the tone of a response and making sure it remains within acceptable parameters, and finding a way to look for patterns in offensive sentence structure. I suspect that the black list of offensive words could be drawn from existing lists of offensive words. This study would ideally be done by reviewing and comparing data from existing studies censoring obscenity produced by chatbots. Should I fail to find sufficient published data, this study could use open-source chat AIs that use machine learning. To elaborate on what responses are considered offensive or obscene, any response that uses excessive profanity or any level of hate speech would be unacceptable.

Source 1: A Reinforcement Learning-based Offensive semantics Censorship System for Chatbots

- Shaokang Cai, Dezhi Han, Dun Li, Zibin Zheng, and Noel Crespi. 2022. A Reinforcement Learning-based Offensive semantics Censorship System for Chatbots. arXiv:2207.10569v1 [cs.CL] 13 Jul 2022. 24 pages. https://arxiv.org/abs/2207.10569
    - This article proposes a type of semantics-based censorship with two components
        - The offensive semantics censorship model, which detects offensive user inputs
        - the semantics purification model, which reduces the offensive responses the AI has learned to produce without using a role back to before that response was learned
    - The term Reinforcement Learning was mentioned several times in this article
    - This article explains the vulnerability that online learning training creates

    - This article cites the several other works it is using as the basis for the proposed method of semantics-based censorship

This article introduces and then explains a two-part semantics-based censorship method. The article cites the many works this method builds upon, which could be helpful sources for this research project. Furthermore, the article also includes and explains the equations used in both parts of the proposed semantics-based censorship method. Near the end of the article is an explanation of the tests used to prove this method's effectiveness and the results of the aforementioned tests. For the sake of my research, the latter of the two parts of the censorship methods is more relevant to my work.

Source 2: Unsupervised Learning by Probabilistic Latent Semantic Analysis

- Thomas Hofmann and Douglas Fisher. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, 42, 177–196, 2001 © 2001 Kluwer Academic Publishers. Manufactured in The Netherlands. 20 pages. https://link.springer.com/article/10.1023/A:1007617005950
    - This article was released in 2001
    - The method featured in this article follows the statistically oriented approach to AIs learning language as opposed to the traditional linguistics school.
    - This article contains the equations needed for its proposed method as well as brief explanations of their meaning
    - The article broaches the topic of methods to avoid overfitting on large data sets
    - Helpfully the article includes the results of testing Probabilistic Latent Semantic Analysis against regular Latent Semantic Analysis.

Although I have some concerns about the considerable age of the article, I do feel like its subject matter could be useful information. My research on the problem of detecting if a chatbot's response is offensive has led me to believe that finding a method of analyzing the semantics of the sentence will be the biggest hurdle. Thus readings like this one that focus on semantic analysis will be crucial to making any headway.

Source 3: Data Curation at Scale: The Data Tamer System

- Michael Stonebraker, Daniel Bruckner, Ihab F. Ilyas, George Beskales, Mitch Cherniack, Stan Zdonik, Alexander Pagan, and Shan Xu. 2013. Data Curation at Scale: The Data Tamer System. CIDR 2013. 6th Biennial Conference on Innovative Data Systems Research (CIDR '13) January 6-9, 2013, Asilomar, California, USA. 10 pages. https://cs.brown.edu/courses/csci2270/archives/2017/papers/data-tamer.pdf
    - This paper outlines a proposed method of automatic data curation referred to as the "Data Tamer"
        - The aforementioned system is an end-to-end curation system we have built at M.I.T. Brandeis
    - data curation is the act of discovering a data source(s) of interest, cleaning and transforming the new data, semantically integrating it with other local data sources, and deduplicating the resulting composite
    - Data Tamer uses standard dictionaries to define certain facts when the DTA deems them relevant to the data being curated
    - Data Tamer allows for human intervention to be requested when the program is too uncertain of its conclusion

Although more directly relevant to the dropped third project pitch, many of the ideas in this article still have merit for pitch A. The style of limited human intervention during program uncertainty is well worth considering adapting for my own uses. Furthermore, the process of data curation involves some degree of semantic analysis, a process which is quickly becoming a focus of my research.

Source 4: Adaptive Clustering of Robust Semantic Representations for Adversarial Image Purification

- Samuel Henrique Silva, Arun Das, Ian Scarff, and Peyman Najafirad. 2021. Adaptive Clustering of Robust Semantic Representations for Adversarial Image Purification. Secure AI & Autonomy Laboratory, arXiv:2104.02155v2 [cs.CV] 7 Apr 2021. 11 pages. https://arxiv.org/abs/2104.02155
    - This article primarily focuses on keeping out bad data from training with images rather than text

    - This article provides the equations used in the method the authors developed, as well as an explanation of what they mean
        - This article does a better job visualizing its equations than some others I have read for my research
    - The software architecture for the proposed method is provided on page 3
    - There is a chart showing the different stages an image can be at in this algorithm at the start of page 6
    - One of the more interesting aspects of this article's method is that it extracts the still usable parts of bad data whilst at the same time learning what parts of the data indicate it is an attack
    - The article ends with data on how well the method being proposed performed during tests

This article focuses on methods similar to what I had in mind but, unfortunately, is for semantic analysis of images, not sentences. Still, some of the equations would likely be similar and may require only slight adjustments. Additionally, one of the aspects of this method that caught my eye is that it was designed to take the parts of an image structure that should still be trained in and use the parts that caused it to be flagged as bad in the first place to learn further what a bad image is. I am not sure how well this can adapt to sentences, but I am intrigued nonetheless.

Source 5: An Analytical Study and Review of open Source Chatbot framework, RASA

- Rakesh Kumar Sharma and Manoj Joshi. 2020. An Analytical Study and Review of open Source Chatbot framework, RASA. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS060723 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 9 Issue 06, June-2020. 4 pages. https://www.academia.edu/download/63825165/an-analytical-study-and-review-of-open-IJERTV9IS06072320200704-117508-1fti3xt.pdf
    - This article explains that RASA was mainly built to answer FAQs using natural language generation
    - RASA's chatbot core has an open-source license

    - One of the design goals for RSA was to make it easy to work with for users unfamiliar with the inner workings of chatbots
        - Lacerda used the core of rasa and presented a new software stack called as Rasa-ptbr-boilerplate for the non-specialist who doesn't know much about the internals of the chatbot
    - RASA supports supervised learning which unfortunately is not quite the unsupervised learning this project centers around, improving
    - RASA features API and Database support
        - The database importing support should come in handy with large data sets

This chatbot core stands out to me for several reasons; first and foremost, its open-source nature makes it much more accessible, something necessary to account for with my limited resources. Secondly, its new user-friendly nature appeals to me as I am far from experienced with modifying chatbots. This does, however, raise the concerns that this design goal of not making users delve to deeply into the inner workings of the core may make it harder to modify the core itself. If the developers never intended for users to modify that part of the program, they may not have bothered making it easy to modify.

Source 6: An intelligent Chatbot using deep learning with Bidirectional RNN and attention model

- Manyu Dhyani and Rajiv Kumar. 2020. An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Materials Today: Proceedings 34 (2021)817-824. G. L. Bajaj Institute of Technology and Management, Greater Noida, Uttar Pradesh, India. 8 pages. https://www.sciencedirect.com/science/article/pii/S221478532034030X
    - This article presents a chatbot designed to be a generalist at first before being trained to excel at a specific task
    - The equations for the attention model are provided at the end of page 3
    - The article features a table analyzing how effective the chatbot was
        - This table has rows for perplexity, learning rate, bleu score, and average time (per 1000 steps)
        - I believe I should look up formal definitions for these terms at some point

    - The article makes mention of a test file written with the express purpose of "validating chatbot's reply"
        - A tool like this tailored to seeing if responses from my own bot are acceptable would worth investing in

This source not only offers up another chatbot platform to work with but also gives insight into how the AI training process is typically conducted. Furthermore, the terms for evaluating chatbot effectiveness will likely become relevant as my research continues. Some of the methods mentioned for validating chatbot answers also piqued my interest.

Pitch B: Another way to study machine learning would be to try and find the most effective way to parse the data on screen into information about a task that an AI could use to complete that task. For the sake of this study, I would use an open-source implementation of Tetris as the novel task. Tetris is a suitable task for this study as it is both simple and includes a random element that will prevent the AI from simply finding an optimal series of keys to press rather than finding a method to play the game. An intuitive way to conduct this study would be to find a suitable open-source machine learning AI and refine methods to turn the current screen into workable data that the AI can understand. However, it would be more effective to gather the needed data from already published methods of automatically parsing the screen into usable data. For that reason, I will seek out published studies that could provide all the needed data or at least insight into how to begin the process of parsing screen data.

Source 1: The Game of Tetris in Machine Learning

- Simon Algorta and Özgür Şimşek. 2019. The game of Tetris in machine learning. arXiv:1905.01652v2 [cs.LG] 10 May 2019. 7 pages. https://arxiv.org/abs/1905.01652
    - Notably, this article has sections dedicated to algorithms and features, the structure of the decision environment, and open challenges.
    - This article covers handcrafted controllers, genetic algorithms, and reinforcement learning and how they have been used in Tetris machine learning
    - This document clarifies what implementation of Tetris is typically used for this line of research

    - This article cites many previous attempts at conquering Tetris using machine learning

This article describes the history of notable and documented machine learning AI built to play Tetris. More critically, this document describes the implementation of Tetris used and the state features given to each AI. This knowledge of how previous AI have been designed and how successful they were will likely be key to knowing where to start my research. Furthermore, this article mentions that a good way to reduce game length and, thus, research time is to reduce the size of the playing space so a game over will occur sooner.

Source 2: Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris

- Patrick Jermann, Marc-Antoine Nüssli, Weifeng Li. 2010. Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris. © 2010, the Authors. 9 pages. https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/HCI2010.7
    - This research project used eye tracking on a pair of Tetris players
    - Sadly this article is more focused on the human element than the machine learning
    - The article defined three significant things for a player to look at
        - Their block
        - Their partner's block (irrelevant to my own work)
        - And the contour of the stack
        - It may be worth getting the AI to focus on only the block and the contour and disregard the rest of the screen
    - The equations for the machine learning used in the research is included near the end of the document in the section Automatic recognition of pair composition
        - Sadly the machine learning aspect was focused on understanding patterns in player interaction, not how they played the game successfully
    - Somewhat interestingly, the article also notes patterns in the parts of the game focused on by more successful players

Although not very directly applicable to my project, there are some potentially essential bits of information to be gleaned. The article breaks down what parts of the screen were most relevant to players and, thus, what parts would be most relevant to an AI given only the current game

screen as input. It also broke down what moves the players had available and when they tended to use them most often. Lastly, this article contains observations about what movement patterns and focuses were most common among more experienced players. Although much of it will have to be adapted back to a single-player game of Tetris, the information in this document has its uses.

Source 3: General Video Game Playing

- John Levine, Clare Bates Congdon, Marc Ebner, Graham Kendall, Simon M. Lucas, Risto Miikkulainen, Tom Schaul, Tommy Thompson, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius. 2013. General video game playing. Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, Volume 6, ISBN 978-3-939897-62-0.,,; pp. 77–83 Dagstuhl Publishing Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany . 7 pages. https://strathprints.strath.ac.uk/57217/
    - a programming language called GDL or Game Description Language mentioned in this article could be a helpful lead
        - That same paragraph also mentions Planning Domain Definition Language
    - This article cites a more formal definition of what a game is, this could be a helpful paradigm to build around for this project.
    - Significantly this article focuses on games with continuous activity, like Tetris, and not just turn-based games
    - This article states that the two main approaches to building a general game-playing AI is deliberative reasoning (e.g. minimax search) or policy learning (e.g. reinforcement learning).
    - The article makes mention of an Arcade Learning Environment (ALE), based on the games for the classic Atari 2600 games console, near the end of page 4
    - The two main AI approaches to investigating a new game are based on reinforcement learning and Monte Carlo Tree Search
    - This article mentions a companion paper that presents Video Game Description Language definitions for three diverse arcade games

This article seems like a potential gold mine of useful sources and ideas. The mentioned companion articles and programming languages tailored to describe the game state to an AI seem to be very promising leads. Furthermore, the style of AI that works with a "first-person perspective" on the game sounds similar to what I had imagined. Looking more into what this style entails, or better yet, finding examples of AI built for this proposed competition, would be a wise investment of time. Needless to say, I have some new leads for my research after this article.

Source 4: Towards a Video Game Description Language

- Marc Ebner, John Levine, Simon M. Lucas, Tom Schaul, Tommy Thompson, Julian Togelius, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius. 2013. Towards a video game description language. Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, Volume 6, ISBN 978-3-939897-62-0.; pp. 85–100 Dagstuhl Publishing Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany . 16 pages. https://strathprints.strath.ac.uk/45278/
    - This is the companion article to the previous article, general video game playing
    - This article mentions several other languages that have similar goals
    - On page 4, the article elaborates on the criteria it considers necessary for VGDL to be a success
    - On page 5, the article breaks down the aspects of the game the language must represent
    - Starting at page 9, the article breaks down how space invaders, lunar lander, and frogger would be implemented in VGDL
    - the article mentions that they have a working prototype built in Python that utilizes Pygame
        - The article ends with a statement that they will continue work on this prototype
    - The article included the code for implementing space invaders using the prototype VGDL language

Although it seems like the goal of the language has shifted from describing the game to AI to quickly making testing environments for AI to interact with the language itself, it could still be a

big help. For one, if the language was finished or at least significantly improved by this point, then there is the possibility that someone has already made a Tetris implementation of this language. Additionally, if Tetris has already been implemented in this language, it may very well have been used in the general game-playing competition; if so, the work of contestants in that competition could be very relevant to my own research.

Source 5: A Video Game Description Language for Model-based or Interactive Learning

- Tom Schaul. 2013. A video game description language for model-based or interactive learning. Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, 6 . Dagstuhl Publishing, Wadern, pp. 85-100. ISBN 9783939897620. 8 pages. https://strathprints.strath.ac.uk/45278/
    - The author of this article is one of the authors from Towards a Video Game Description Language, Tom Schaul
    - This language was developed to generate diverse benchmark problems for learning and planning algorithms
    - This article contains much more comprehensive examples of VGDL syntax then the previous article, which was written when VGDL was much less developed
    - This article covers what the interface would look like for humans and, much more importantly, nonhuman players
    - The name of the language in this article is now PyVGDL, as it is essentially just a much higher-level version of PyGame written with the specifications of VGDL
    - Bots using this language interact with the game by taking one action per cycle
        - The bot has access to the getSensors and performAction methods

This language designed to be easily written and make games easy for AI to play could prove to be very helpful. This article states the PyVGDL supports continuous physics and object spawning, which is good if I plan to make or find an implementation of Tetris written in this language. A potential problem I have noticed is that no mention has been made yet of whether or not PyVGDL supports players or objects that take up more than one tile on the map.

Source 6: XML-Based Video Game Description Language

- Jorge R. Quiñones and Antonio J. Fernádez-Leiva. 2019. XML-based video game description language. Volume 8, 2020. Digital Object Identifier 10.1109/ACCESS.2019.2962969. 14 pages. https://ieeexplore.ieee.org/abstract/document/8945249/
    - This article is another video game description language based in XML instead of Pygame
    - This article mentions other VGDLs in section 2 and then ends the article by comparing itself to other VGDLs
        - This comparison includes a chart of which features are available in which VGDLs
        - The author believes that XVGDL is an improvement over the previously read about PyVGDL
    - This language is set up to allow other engines and tools to be used
    - When defining the player attributes, one can instead assign those attributes to a specified AI
        - This is set up to support externally implemented AI
        - AI can be assigned to NPCs as well
    - All the work on this VGDL is available on GitHub
    - The article ends with a full implementation of Pac-Man as an example of all the features previously mentioned being used.

I believe that with this more recent article, I have found a suitable environment for testing. The built-in support for importing other tools/extensions and the focus on AI support seem obvious advantages for my work. Going forward, I focus any reading on general game-playing AI themselves instead of the environment to test them in.