# Synthesizing controversial sentences for testing the brain-predictivity of language models

by

Lara I. Rakocevic

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2021



Author .....
Department of Electrical Engineering and Computer Science
Jan 15, 2021

Certified by .....
Evelina Fedorenko
Associate Professor
Thesis Supervisor

Certified by .....
Noga Zaslavsky
BCS Fellow in Computation
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

# Synthesizing controversial sentences for testing the brain-predictivity of language models

by

Lara I. Rakocevic

Submitted to the Department of Electrical Engineering and Computer Science
on Jan 15, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

## Abstract

Recent research has seen the rise of powerful neural-network language models that are sufficiently computationally precise and neurally plausible as to serve as a jumping-off base for our understanding of language processing in the brain. Because these models have been developed for optimizing a similar objective (word prediction), their brain predictions are often correlated, even though the models differ along several architectural and conceptual features, yielding a major challenge for testing which model features are most relevant for predicting language processing in the brain. Here, we address this challenge by synthesizing new sentence stimuli that maximally expose the disagreement between the predictions of a set of language models ('controversial stimuli'), which would not naturally occur in large language corpora . To do so, we develop a platform for systematizing this sentence synthesis process, providing a way to test different model-based hypotheses easily and efficiently. An initial exploration with this platform has begun to give us some intuition for how choosing from different pools of candidate words affect the kinds of sentences produced, and what kinds of changes tend to produce controversial sentences. For example, we show that the disagreement score, or the maximum amount of disagreement between models for a sentence, converges. This approach will eventually allow us to determine which models perform in the most human-like way and are most successful in predicting language processing in the brain, thus hopefully leading to insights on the mechanisms of human language understanding.

Thesis Supervisor: Evelina Fedorenko
Title: Associate Professor

Thesis Supervisor: Noga Zaslavsky
Title: BCS Fellow in Computation

# Acknowledgments

Over the course of this last year, the world around us has been upended, turned topsy-turvy, and often produced more questions than it has answered. Amidst this crazy new reality, I began and eventually finished my thesis. While the world has been unpredictable, what's been consistent is my gratitude to the people in my life who have continued to inspire and motivate me through the duration of this process. It is only through the guidance and excessive kindness of those around me that I was able to complete this journey, and I am eternally grateful for that.

I would like to thank Ev Fedorenko and Noga Zaslavsky, my advisors on this thesis. To Ev, thank you for taking me on and introducing me to this rich field of problems. Talking with you, seeing your passion and expertise, has inspired me both within the scope of this project and beyond, and I am so excited to have you as a role model as I explore future directions. To Noga, your constant guidance and engagement with every question I had, was crucial to my learning process. I was able to iterate on my mistakes and understand why they were mistakes in a much more focused and practical way, due to your thorough and thoughtful explanations and nudging in the right direction. To you both, it is incredibly empowering to see such strong women in STEM, and it is truly an inspiration to have you as role models ahead of me.

In addition, I'd like to thank my sibling Ines for their inspiring brilliance, their constant love, and their poignant understanding of how to be a good human being. These last few months have been trying for us all, and I know you've gone through your fair share of struggles, but without fail you have been a light in my life and brought inexplicable joy even in the bleakest of moments.

To my parents Ana and Vlad, I'd first like to say, in the most loving way possible: told you so! I owe the two of you so much that I won't ever be able to repay or explain. One thing that I can easily say though, is that I have always looked to you as a source of drive and ambition, and that I could not have achieved any of my successes (and likely some of my failures too) without you. Thank you for your constant vigilance and for instilling in me a sense of accountability and intention without which I would

be lost.

Finally, I'd like to give a shout out to my lovely friends who kept me from getting cabin-fever over the course of this pandemic. Countless video chat sessions, phone calls, and essay-like texts kept me sane over these last few months, which not only helped me to finish my thesis, but pushed me to think more deeply about my motivations, my convictions, and my values, for which I will always be thankful.

# Contents

# List of Figures

# Chapter 1

# Introduction

Language is our signature human cognitive skill and the cornerstone of human culture and civilization. It underlies human communication, which has allowed us to achieve the unimaginable; from delving into the minutiae of our own being, to developing an ever more nuanced understanding of the universe and its mysteries. This amazing and uniquely human skill has been explored from all directions, from philosophy to neuroscience, and these studies have given us a rich understanding of the mechanisms of language comprehension and production. However, many questions persist. As we gain new tools across many different disciplines, our avenues of exploring these questions widen, and we can use these techniques to help unveil mysteries that seemed inexplicable as little as a few years ago. The innovation of computationally precise and neurally plausible models of language processing allows for a new direction of research; these models can serve as quantitative hypotheses for how core aspects of language might be implemented in neuronal circuits. In this project, I aim to make use of such models to help elucidate human language cognition and function.

## 1.1 Statement of the Problem

Understanding the human brain remains one of the greatest scientific challenges. Especially hard to characterize are mechanisms that support human-unique abilities, like language. In other domains, like vision, which is similar between humans and

other mammals, animal models have proven invaluable. In language and other high-level domains, it has been difficult to get to mechanistic-level accounts of the relevant mental processes. However, recent advances in artificial intelligence and machine learning stand a chance to change this. In particular, state-of-the art artificial neural network (ANN) language models now achieve incredible performance on many complex language tasks, and thus can serve as computationally precise hypotheses for how the human mind and brain may solve the problem of language understanding. Indeed, recent attempts to relate human neural data during language processing to representations extracted from ANN language models have shown a lot of promise. While exceptions exist [3], most model-to-brain fit evaluations so far have been carried out using existing neural datasets. Although these datasets are useful for initial evaluations of which model achieves the best fit to human neural recordings, they are not perfectly suited to answer why some models may provide a better fit than others.

To answer this question, one must be able to isolate the core properties of the chosen models. The next frontier is then to build and test ANN language models that minimally differ in architectural or training features, in a way that embodies distinct cognitive hypotheses about the computations executed in the language network and collect new language comprehension data using materials that are optimized for discriminating among different models. And a core first step in this process — the step I undertake in this thesis— is to develop methods for creating stimulus sets that will tease apart the differences in models in such a way that the behavior across models is unique enough to truly distinguish the distinctive attributes arising from each respective architecture.

Creating such a set of stimuli in the domain of natural language provides is challenging. Though work of a similar nature has been done in the context of vision, the methods used for producing such stimuli in vision are not easily transferable to our use case. In addition, while others have worked on producing adversarial examples in natural language, examples that cause a model to make a mistake, a method for producing controversial examples, examples that elucidate differences between models as described above, has not been developed. We use work that has been done with

controversial examples in vision as an illustrative example for our work here, and extend previous work in producing adversarial examples to producing controversial examples.

## 1.2 Preliminaries

### 1.2.1 ECoG

Electrocorticography (ECoG) is a type of electrophysiological monitoring that uses electrodes placed directly on the exposed surface of the brain to record electrical activity from the cerebral cortex.

### 1.2.2 fMRI

fMRI, or functional magnetic resonance imaging or functional MRI, is a technique that measures brain activity by detecting changes associated with blood flow. Cerebral blood flow and neuronal activation are coupled; when an area of the brain is in use, blood flow to that region increases.

### 1.2.3 Language Model

A language model is a probability distribution over sequences of words. Given a sequence of length $k$, a language model assigns a probability $P(w_1, \ldots, w_k)$ to the whole sequence. Such models can also assign probabilities to each individual word at each point in a sentence.

### 1.2.4 RNN

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. Their architecture is shown below.

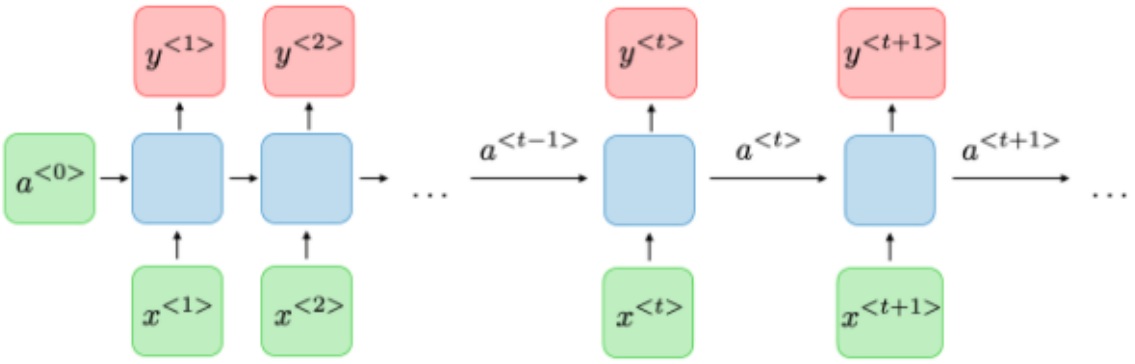


Figure 1-1: RNN Architecture

For each timestep $t$, the activation $a^{<t>}$ and output $y^{<t>}$ can be expressed with the following equations:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are all coefficients, and $g_1, g_2$ are activation functions.

### 1.2.5 Backpropagation

Backpropagation is an algorithm of supervised learning for artificial neural networks using gradient descent. Backpropogation computes the gradient of the loss function with respect to the weights of the network in order to train neural networks. Each of the neural network's weights gets updated proportional to the partial derivative of the loss function in each iteration of training.

### 1.2.6 LSTM

Long short-term memory (LSTM) is a specific recurrent neural network (RNN) architecture that is capable of maintaining long-term dependencies. During backpropogation, classical RNNs suffer from the vanishing gradient problem - when a gradient becomes so small that it no longer has any effect on the weight's value. LSTMs have

internal mechanisms called gates that can regulate the flow of information as well as a cell state, allowing them to pass down relevant information. The cell remembers values over varying time intervals, while the gates (input, output, and forget gate) regulate the flow of information in and out of the cell.

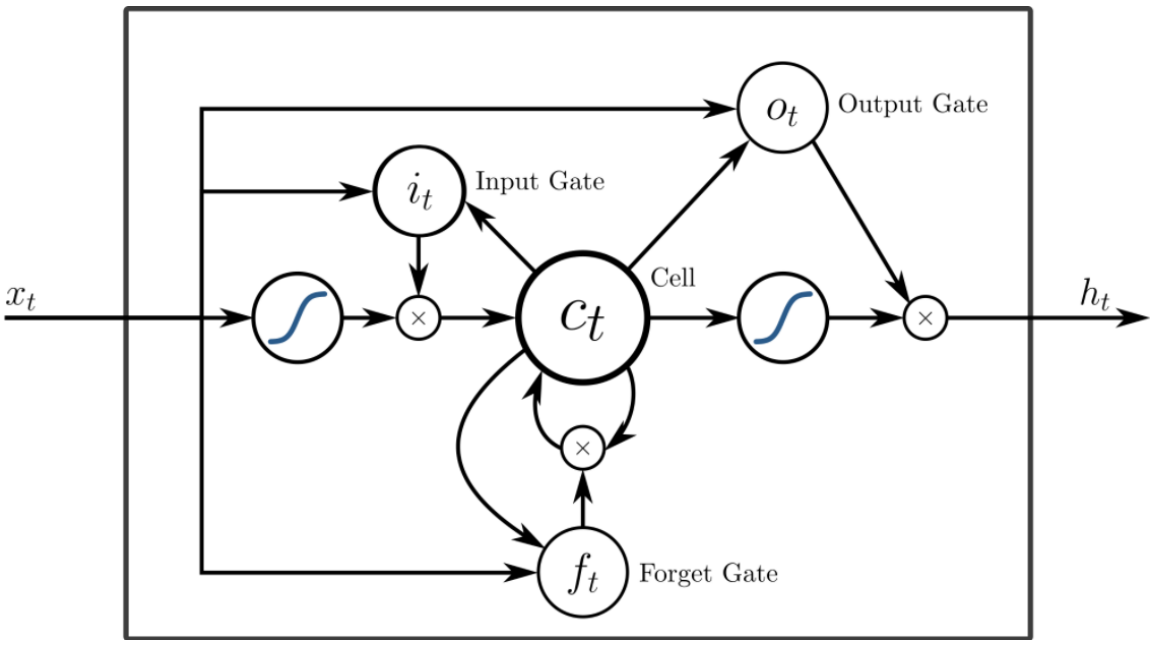


Figure 1-2: LSTM Cell Architecture

### 1.2.7 Transformer

Sequence-to-sequence (seq2seq) models in NLP are used to convert sequences of one type to sequences of a different type. Transformers are models that aim to solve seq2seq tasks while handling long-range dependencies with ease. It was first introduced by Vaswani et al in their 2017 paper "Attention Is All You Need". Transformers transform one sequence into another using an Encoder, Decoder, and Attention mechanisms. The architecture proposed in the paper is shown below. [20]

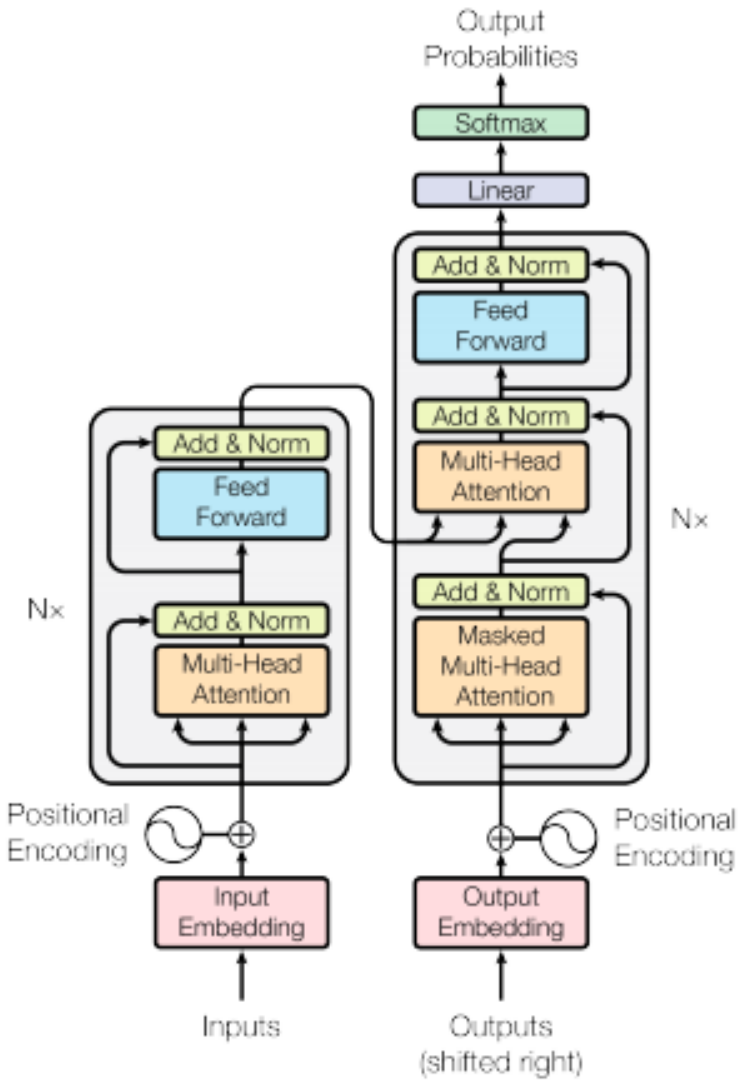


Figure 1-3: Figure from Vaswani at al showing the architecture of a Transformer

## 1.3 Related Works

### 1.3.1 Neural Networks to Model Brain Behavior

Recent research in natural language has seen the development of computationally precise and neurally plausible models of language processing. This link between mind and machine has been explored in depth and seen much success in vision [12], [17], [22], [25], [9]. Of great help in this work has been the use of animal models.

For example, Yamins et al (2013) sought to construct models of the ventral stream, a hierarchical cortical neuronal network used to solve object recognition tasks. Using a novel optimization procedure for category-level object recognition problems, they were able to construct models of the ventral stream. New developments at the time had bolstered the previous long-standing hypothesis that the visual input captured by the retina is rapidly processed through the ventral stream into an effective, "invariant" representation of object shape and identity, by showing that the abstract category-level visual information is accessible in IT (inferotemporal) cortex, but much less effectively in lower cortical areas. The idea was that in developing encoding models that map the stimulus to the neural response of visual area IT in macaque monkeys, some understanding of object recognition in humans would come to light. This model helped further develop a long-standing functional hypothesis that the ventral stream is a hierarchically arranged series of processing sages optimized for visual recognition. [21] Finally, this demonstrated that the activations of a convolutional neural network trained on ImageNet in response to natural images could predict activations in a macaque monkey's visual cortex in response to the same images.

In work published a year later also by Yamins et al, it was found that models that performed better on an object categorization task were also more likely to produce outputs that aligned more closely to IT neural responses. Additionally, when optimizing for performance, the best-performing models predicted neural output as well as those models directly selected for neural predictivity, although the reverse is not true. Together, these results imply that performance optimization is an efficient means to identify regions in parameter space containing IT-like models. [22] Shrimpf et al extended on this work in 2018 by demonstrating that this performance-driven approach holds in a broad sense when evaluated on multiple deep neural networks in a wide range of ImageNet performance regimes, but fails to produce a network exactly matching the brain when reaching human performance levels. [17]

These methods have even been extended to the domain of human auditory behavior. It has been found that a task-optimized neural network can replicate human auditory behavior and predict brain responses. The network performed auditory

tasks as well as humans and exhibited human-like errors despite not being optimized to do so, suggesting common constraints on network and human performance. The network additionally predicted fMRI voxel responses substantially better than traditional spectrotemporal filter models throughout auditory cortex. [12]

Progress has been made in several iterations in the field of natural language. First, distributional word representations were found to be useful in predicting human brain activations, when subjects were presented with single words [13]. Then, this result was extended by Huth et al using distributed word representations, and again by Pereira et al to sentence stimuli. All this has led up to work by Shrimpf et al (2020) to find that certain models could predict language processing in the brain.

To expand on these studies, first Mitchell et al., (2008) demonstrated that distributional word representations could be used to predict human brain activations. In their paper, they present a computational model that predicts the functional magnetic resonance imaging (fMRI) neural activation associated with words for which fMRI data had not yet been collected. The model was trained with data from a trillion-word text corpus as well as observed fMRI data associated with viewing several dozens of concrete nouns. Once trained, the model would predict fMRI activations for thousands of other concrete nouns from the text corpus, yielding a highly significant accuracy over the nouns for which the experimenters did have fMRI data available for.[13]

The mechanics of the model were as follows. Given an arbitrary stimulus word $w$, the first step gets frequencies of co-occurence with a set of common 20 verbs for each noun, and those are the intermediate semantic features. In the next step, the neural fMRI activation would be predicted at every voxel location in the brain, as a weighted sum of neural activations contributed by each of the intermediate semantic features. For context, the voxel is a 3-dimensional unit that embeds the signals in brain scans. As the MRI machine scans through each dimension of the brain millimeter by millimeter, voxels are formed to enclose the signals created by protons-magnet interactions. In particular, the predicted activation $y_v$ at a voxel $v$ in the brain for a word w can be calculated in the following way: $y_v = \sum_{i=1}^{n} c_{vi} * f_i(w)$ where $f_i(w)$ is the value of the $i^{th}$ intermediate semantic feature for a word $w$, $n$ is the number

of semantic features in the model, and $c_{vi}$ is a learned scalar parameter that specifies the degree to which the $i^{th}$ intermediate semantic feature activates voxel $v$. This essentially allows us to predict the full fMRI image across all voxels for a stimulus word $w$ as a weighted s of images, one per semantic feature $f_i$. These semantic feature images, defined by the learned $c_{vi}$, constitute a basis set of component images that model the brain activation associated with different semantic components of the input stimulus words.[13]

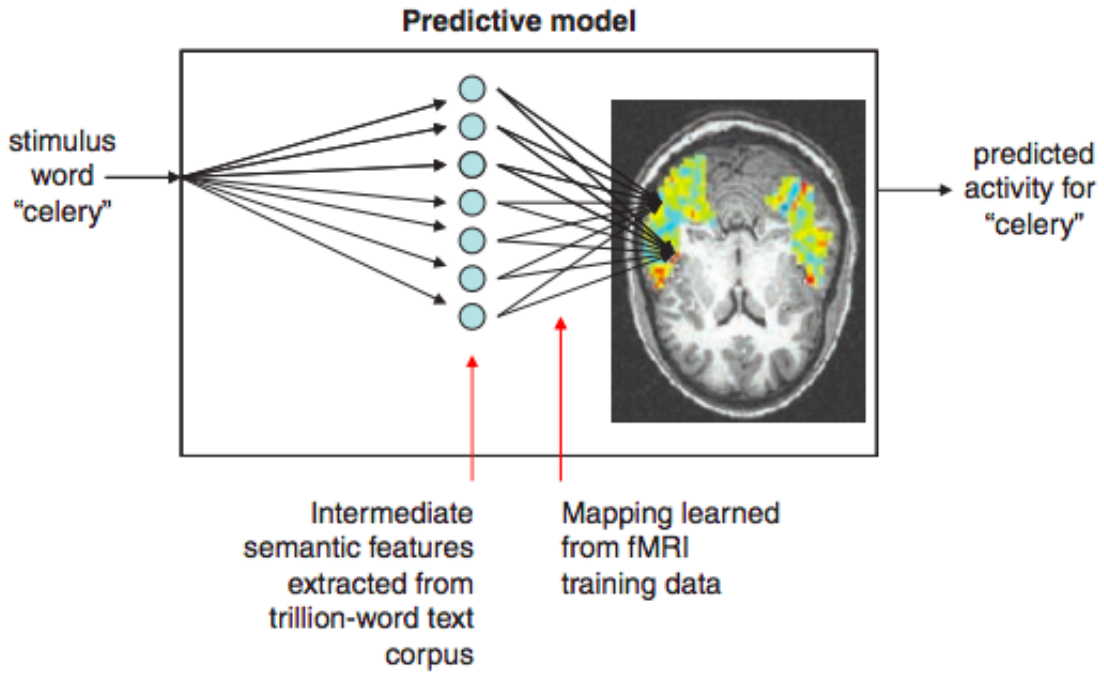


Figure 1-4: A figure from Mitchell at al showing the steps of the predictive model

In Pereira et al's 2018 work, they present a new approach for building a brain decoding system in which words and sentences are represented as vectors in a semantic space constructed from massive text corpora. By efficiently sampling this space to select training stimuli shown to subjects, they were able to maximize the ability to generalize to new meanings from limited imaging data. Prior to this study, a number of studies had attempted to learn a mapping between particular semantic dimensions and patterns of brain activation. The idea behind this is that if such a mapping can make predictions about neural responses to new stimuli, this means the

underlying model must have captured some aspect of the representation of meaning. Previous studies in the natural language domain had been restricted to relatively constrained sets of stimuli (like concrete nouns, seen above), which left the question of whether the models would generalize to meanings beyond those that they were built to accommodate open. In this paper, the authors introduce an approach for building a universal brain decoder that can infer the meanings of words, phrases, or sentences from patterns of brain activation after being trained on a limited amount of imaging data. The goal behind this was to develop a system that would work on imaging data collected while a subject read natural linguistic stimuli on any topic, including abstract ideas. In addition to providing an excellent and exciting new approach for building a universal brain decoder, this paper produced an exceptional set of brain recordings that paved the way for future analyses and experiments. [14]

Finally, research has been done in the Fedorenko lab concerning whether state-of-the-art artificial neural network language models capture human brain activity elicited during language comprehension using a pipeline from vision research [17]. 43 language models were tested on three neural datasets (fMRI [14] and ECoG [5]), spanning all major classes of existing language models and included embedding models, recurrent neural networks, and variants of attention based architectures—transformers. The results showed that the most powerful transformer artificial neural network models could accurately predict neural responses. This shows that specific language models can capture substantial variance in neural responses to language, which provides the first computationally precise account of how the human brain may solve the problem of language comprehension. [16]

To compare a given model to a given dataset [5, 14], the authors presented the same stimuli to the model that were presented to humans in neural recording experiments and 'recorded' the model's internal activations. To retrieve model representations, they treated each model as an experimental participant and ran the same experiment on it that was run on humans, including adjusting how words, sentences, or paragraphs were input in order to mimic the human experiment properly for each type of model. After the processing of each word, they retrieved model representations

at every computational block (i.e. an LSTM cell). When comparing against human recordings spanning more than one word such as a sentence [14], they additionally aggregated model representations.

This body of work has seen particular success over the last few years, and the link between mind and machine becomes ever more attainable. With the rise of increasingly accurate models of human cognitive functions, it becomes critical to synthesize stimuli for probing and understanding the underlying functions of such models.

### 1.3.2 Synthesizing Stimuli for testing Neural Networks

Deep neural networks are powerful learning models that have achieved excellent performance on a wide array of problems in the domain of image recognition problems. This high performance is possible because they can express arbitrary computation that consists of a modest number of massively parallel nonlinear steps. Since the resulting computation is automatically discovered by backpropagation via supervised learning, it can be difficult to interpret and can have counter-intuitive properties. In their 2014 work, Szgedy et al explore the intriguing properties of neural networks, including the semantic meaning of individual units as well as the stability of neural networks with respect to small perturbations to their inputs. [19]

This exploration into these perturbations laid the basis for developing adversarial examples. The authors expected that a state-of-the-art deep neural network that generalizes well on object recognition tasks would be robust to small perturbations of the input, given that these small perturbations cannot change the object category of the input image. However, they found that by applying imperceptible non-random perturbation to a test image, it's possible to arbitrarily change the network's prediction. These perturbations were found by optimizing the input to maximize prediction error.

In addition to finding that neural nets were vulnerable to these perturbations, which yielded "adversarial examples", they made three interesting observations. The first was that the adversarial examples that they managed to generate were virtually

impossible to distinguish from each other by the human eye, yet were misclassified by the original network. The second was that a large fraction of the adversarial examples were misclassified by networks trained from scratch with different hyperparameters. Finally, they found that a large fraction of the examples were misclassified by networks trained from scratch on a disjoint training set. From this, they concluded that adversarial examples seemed to be somewhat universal, instead of just being the result of overfitting to a particular model or selection of the training set.

Building on this work, Goodfellow et al attempted to understand why these adversarial examples might function in the way they do. They officially defined adversarial examples to be "inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence", and claimed that the primary cause of a neural network's vulnerability to adversarial perturbation is the linear nature of the network. Through their work, they came to the conclusion that instead of learning the true underlying concepts that determine the correct output label, the algorithm builds a fraudulent understanding that works for naturally occurring data, but is exposed as a fake when one visits points in space that do not have high probability in the data distribution. [7]

While this question of adversarial examples and perturbations had been explored in great depth in the domain of vision, there was little work done exploring the potentials of such examples in natural language initially. This was plausibly for several reasons. While in the image domain, these perturbations are often virtually indistinguishable to human perception, in the natural language domain, small perturbations would be clearly perceptible. Even the replacement of a single word could drastically alter the semantics and structure/grammaticality of a sentence.

In addition, the methods used to generate adversarial examples could not easily transfer. Adversarial examples in the image domain had been generated through solving an optimization problem, attempting to induce misclassification while minimizing the perceptual distortion. In response to the computational cost of such approaches, fast methods were introduced which, either in onestep or iteratively, shift all pixels

simultaneously until a distortion constraint was reached. Nearly all such methods are gradient based. [23]

All these methods, however, rely on the fact that adding small perturbations to many pixels in the image will not have a noticeable effect on a human viewer. Any perturbations that could be added to a sentence, however, would be immediately noticeable. Additionally, while image pixel values are continuous, words in a sentence are discrete, so we cannot compute the gradient of the network loss function with respect to input words. One possibility to create an analogous computation would be to project input sentences into continuous spaces, then consider this as a model input. This approach still fails, though, because it assumes that replacing every word with words nearby in the embedding space will not be noticeable. In the language domain, though, besides semantic cohesiveness, syntactic coherence is also a factor, and without taking this into account, improper sentences will arise that clearly look wrong to a human eye.

While these problems have been explored extensively since, there is continual work being done to improve the production of adversarial examples in natural language processing. Different approaches have been taken to generating such adversarial examples. A few white-box attacks, attacks where attackers have full knowledge about the algorithm, have employed a method adding noise whose direction is the same as the gradient of the cost function with respect to the data (the fast gradient sign method), [7], while others have considered black-box adversarial examples for NLP evaluation [8]. Some have explored adversarial examples on the sentence level, while others have looked to create adversarial examples on the paragraph level, adding distracting but non-contradictory sentences to the input paragraph [8]. Others still have tried to rely on synthetic and naturally occurring language errors to generate noise, which have ultimately successfully been able to fool models even when humans cannot be fooled [2]. Approaches like these have looked at making changes directly on text input space, but other work has been done to perform perturbations in the continuous space that has been trained to produce semantically and syntactically coherent sentences automatically with some success [24].

Among this larger body of work, a paper by Alzantot et al in particular is relevant to this research project. The focus of their research is to generate semantically and syntactically similar adversarial examples that fool well-trained sentiment analysis and textual entailment models. In addition, they were able to validate that the examples are both correctly classified by human evaluators and similar to the original examples via a human study. They do so by using a black-box population-based optimization algorithm. In their approach, they aim to minimize the number of modified words between the original and adversarial examples, but only perform modifications which retain semantic similarity with the original and syntactic coherence. To do so, they develop an attack algorithm that exploits population-based gradient-free optimization via genetic algorithms. While this works, this approach does not use the gradient at all, for the reasons mentioned previously. They note that this may mean that their approach is comparably inefficient. [1] This is in particular relevant to our research due to the systematic and algorithmic nature of their sentence production, which has served as a model for the platform described later in this paper, as well as because of their focus on coherence in their sentences. Rather than hand-crafting sentences or selecting from pre-existing sets, they have worked to synthesize new sentences from existing ones, aiming to increase the adversarial nature of sentences through their algorithm.

Although Alzantot et al. bring the problem of adversarial inputs to the natural language domain, it does not address one of the crucial ideas that had been noted about adversarial examples at large: that they often are shared between models, and can fool multiple models in similar ways. In the current, the objective is not to compare models, so the authors are not trying to optimize for this. Therefore, this work provides a good starting point for the idea of "fooling models" in the natural language domain in some sense, but does not allow us to parse out these differences very clearly.

Instead, a different concept is introduced which can do this much more effectively.

### 1.3.3 Optimal Experiment and Stimulus Design

To maximize the chance for success in an experiment, good experimental design is needed. However, the presence of unique constraints may prevent mapping the experimental scenario onto a classical design. In these cases, we can use optimal design: a powerful, general-purpose tool that offers an attractive alternative to classical design and provides a framework within which to obtain high-quality, statistically grounded designs under nonstandard conditions. It can flexibly accommodate constraints, is connected to statistical quantities of interest and often mimics intuitive classical designs. [18] [11]

Optimal experiment and stimulus design is a long-standing research area. Golan et al joins this rich field with their 2020 paper, exploring the idea of "controversial stimuli", for pitting neural networks against each other as models of human recognition. The authors note that since distinct scientific theories can make similar predictions, in order to judge between theories, experiments must be designed for which the theories make distinct predictions. In this experiment in particular, the authors wish to compare deep neural networks as models of human visual recognitions. [6]

In order to efficiently compare the ability of these various models to predict human responses, it is critical to create a new set of stimuli - controversial stimuli. For a controversial stimulus, each different model produces a distinct response. The reason that one cannot use an adversarial example to do this same task is that adversarial stimuli are not guaranteed to expose differences between different models, because they are not designed to probe the portion of stimulus space where the decisions of different models disagree.

An adversarial example is a stimulus that is controversial between a model and an oracle that defines the true label. By definition, it requires the evaluation of ground truth in the optimization loop. Evaluating ground truth is no trivial task, however, since it may require human judgement. Often as a replacement, adversarial attacks are used as a stand-in to this human judgement component. Controversial stimulus synthesis, on the other hand, allows us to compare two models without needing

to evaluate or approximate ground truth within the optimization loop; only once, once optimization is completed, is the ground truth necessary to determine which model responded correctly. Thus controversial stimuli allow us to use more costly and compelling evaluations of ground truth instead of relying on other measures, as in adversarial examples.

The authors of this paper note that even deep neural network models that exhibit highly human-like responses when tested on in-distribution stimuli often show substantial differences from human responses when tested on out-of-distribution stimuli. An example of this might include images from a different domain, images degraded by noise or distortion, or images that have been adversarially perturbed. Assessing a model's ability to predict human responses to out-of-distribution stimuli provides a truer test of the model's inductive bias, which is the explicit or implicit assumptions that allow the model to generalize from training to novel stimuli. In order to predict human responses to new stimuli, the model has to have an inductive bias similar to that of humans. This is because they require that the model generalize beyond the training distribution.[6]

The final result of the paper was that the authors were able to synthesize controversial stimuli to maximize the disagreement among models which employed different architectures and recognition algorithms, and using this quantified how accurately each model predicted the human judgments.[6]

As mentioned before, this is a long-standing field of research that has been explored along many avenues. While I have discussed a relevant example in vision above, note that important breakthroughs in this field have occurred along various threads of research; the McDermott Lab specifically has done thorough work in exploring optimal design in the context of audition and auditory stimuli.

For example, work done in the McDermott lab has delved into whether contemporary deep neural networks can be used in this way to gain normative insights about complex perceptual tasks. Specifically, they explore the interplay of peripheral coding and stimulus statistics in pitch perception, training artificial neural networks to estimate fundamental frequency from simulated cochlear representations of natu-

ral sounds. The best-performing networks replicated many characteristics of human pitch judgments. In order to figure out how human ears and our environment shape these characteristics, optimized networks were given altered cochlea or sound statistics. Human-like behavior emerged only when cochleae had high temporal fidelity and when models were optimized for natural sounds. [15] Exploring the capabilities of deep neural networks has become an important question, as deep neural networks have become more commonplace as models of sensory systems: how similar are they really to various biological systems? In this vein, Feather et al synthesized model metamers – stimuli that produce the same responses at some stage of a network's representation - for natural stimuli by performing gradient descent on a noise signal, matching the responses of individual layers of image and audio networks to a natural image or speech signal. [4]

## 1.4 Contributions

This thesis fits into a project that has a specific scientific goal: to develop controversial stimuli within the domain of natural language. The goal of this thesis is slightly broader; in this work, our main contribution is a platform that allows us to explore the affects of different parameters on synthesizing controversial stimuli for natural language. This platform gives flexibility in being able to test different combinations of a proposal mechanism, different scoring functions, and different models. This allows a user to test myriad ways of synthesizing controversial sentences, and allows for ease of testing different options and hypothesis. Using this platform, we can explore the moving parts of this synthesis process, and distill down what aspects are most helpful, and add the most controversy to a sentence while retaining the most coherent sentence structures.

By enabling such flexibility, this platform allows us to systematize the process of searching for what aspects of language and sentence creation incite controversy. Instead of simply selecting from a preexisting corpus of sentences, or hand-crafting sentences on an as-needed basis, we have built a platform that allows us to isolate

specific hypotheses and analyze how they affect the sentence synthesis process. This platform has begun to give us an understanding towards what a controversial sentence should look like, what the underlying aspects of a sentence are that make it controversial, and what kinds of methods work for creating such stimuli. We can begin to delve into "blind spots" in the potential space of sentences, and exploit edge cases that will help us understand how our models' inductive biases function with increasing clarity.

In the following chapter, I introduce the algorithm that the platform is based on, along with several relevant abstractions which will be essential to understanding how the platform works and the flexibility it allows us. Then, I will show the algorithm in action in order to give a more intuitive understanding of how the algorithm accomplishes the task of producing controversial sentences.

Finally, I will outline several future directions for this project, including different ways to utilize the platform, and how these controversial sentences will be used once they have been developed.

# Chapter 2

# A framework for synthesizing controversial sentences

In this section, I first outline the desiderata of this project; what we aim to achieve with this platform. From there I introduce the algorithm that the platform is based on, along with several relevant abstractions which will be essential to understanding how the platform works and the flexibility it allows us. Along with each of these abstractions, I include a brief description of the options I have already implemented and tested for the use of synthesizing these controversial sentences. In addition, I discuss additional constraints introduced to the algorithm in order to ensure more coherent sentences.

## 2.1 Desiderata

The problem that this algorithm aims to address is two-fold. The final goal of this algorithm is to create controversial sentences, sentences that will exploit current "blind spots" in the realm of all possible sentences - whether that be naturally occurring sentences or "sentences" consisting of random lists of words. This space of sentences is currently undefined - while we have general notions of what kinds of sentences might be confusing for humans to read, we have little understanding of what kinds of input would incite controversy among models. There are several moving parts

which must come together in order to accurately and efficiently explore this space. By creating a flexible platform that allows us to individually explore the effects of these moving parts, we can gain a better understanding of what kinds of hypotheses really incite controversy, and in turn gain some intuition for what controversy looks like in language, and how we can eventually maximize such a measure.

### 2.1.1 Objective Function

In order to develop a set of controversial sentences, one must first define an idea of controversy, or a way to evaluate a single sentence for how controversial it is. My goal is to capture the difference in the output of multiple models, elicited by a single sentence. Within the task of language modeling, an effective way to do so would be to compare the probability distributions produced by each model over the next word possibilities. To this end, I employ a method that measures the similarity and differences between two or more probability distributions. Ultimately, it is this metric that we are trying to optimize for, thus in the algorithm it is denoted as a gain function, $G$. The parameters to this gain function are a sentence $s$, the model list $L$, and the vocabulary length $V$.

#### Jensen-Shannon Divergence

As noted in previous sections, the final goal of this platform is to both synthesize sentences that have high controversy, and explore why high controversy occurs in these contexts. In both of these goals, the idea of controversy is absolutely key; it is our objective function, what we are trying to numerically maximize, and what we are trying to theoretically understand. Insofar as we are interested in the controversy between the "reactions" of models, and that the "reactions" of models are essentially probability distributions over words, we need a measure that can capture the differences in these probability distributions in such a way that is consistent, clear, and theoretically sound.

### Definition

The Jensen-Shannon divergence ($JSD$) is a divergence measure which is always finite for finite random variables. It quantifies how similar or different two or more distributions are from each other.

The formal definition is as follows:

$$JSD_{\pi_1,\pi_2,...,\pi_n}(P_1, P_2, ..., P_n) = H(\sum_{i=1}^{n} \pi_i P_i) - \sum_{i=1}^{n} \pi_i H(P_i)$$

where $\pi_1, \pi_2, ..., \pi_n$ are weights that are selected for the probability distributions $P_1, P_2, \ldots, P_n$, and $H(P)$ is the Shannon entropy for distribution $P$.

### Justification

The Jensen-Shannon divergence have several properties in particular that make it useful as a theoretical measure of controversy in language. While the Jensen-Shannon divergence is based on the Kullback–Leibler divergence, it has several important differences. Crucially, it is symmetric and always has a finite value. Since we want to explore how different two distributions are from each other, symmetry is essential. Since our theoretical space of changes is infinite, it is indispensable that there is always a finite value with which to measure each change. In addition, the Jensen-Shannon divergence can be generalized to include as many probability distributions as necessary, so one can compare multiple distributions at once instead of just making pairwise calculations. Thus, we can explore how many models' inductive biases contrast. Finally, there is a natural upper limit to the score as well when using the Jensen-Shannon Divergence, thus we actually have a upper limit to use as a sanity check for how much controversy produced is a useful and substantial amont of controversy. For $M$ probability distributions, the upper limit is $ln(M)$. For these reasons, I chose to use the Jensen-Shannon divergence as a metric for the similarities and differences between these probability distributions.

In addition to these theoretical justifications, the Jensen-Shannon divergence has shown to have practical justification as well in this context. In order to validate that

the Jensen-Shannon Divergence is indeed a good proxy for model discriminability on new neural data, we use regression weights from [16] and calculate the cosine distance for novel stimuli. For the same sentences, we found the Jensen-Shannon Divergence as well. Plotted below are the scores of the cosine distance versus Jensen-Shannon divergence for each sentence.

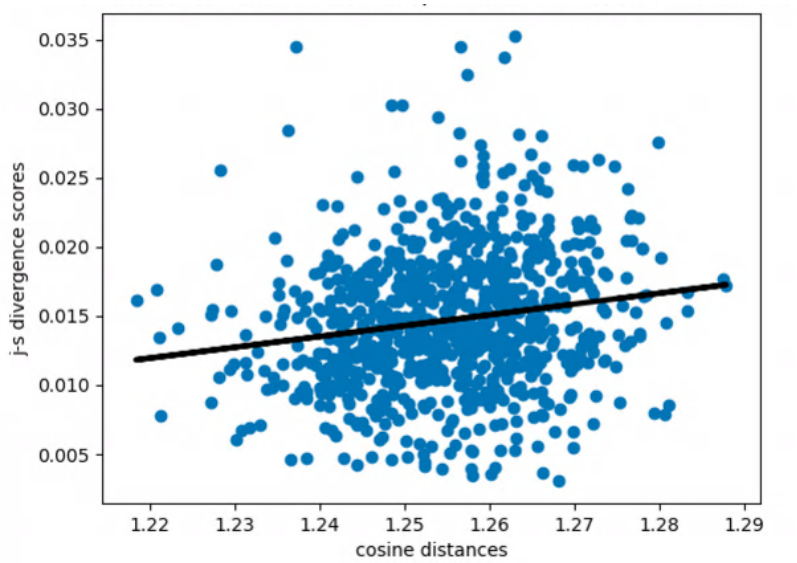


Figure 2-1: Correlation Between Jensen Shannon Divergence and Cosine Distance

For this correlation, $R^2 = 0.02903$, and $p = 3.4575 * 10^{-7}$. This shows that there is a significant, though weak, correlation between the Jensen-Shannon Divergence and Cosine Distance, which is based on the brain activations from the Schrimpf paper. Thus there is a plausible reason to use the Jensen-Shannon divergence as a measure for controversy, knowing that eventually the controversy in these sentences will be used to compare distances between models in the context of human language processing.

## 2.2 Statement of the Algorithm

In this section, we present the algorithm and explain how it works. The high level idea behind the algorithm is to take a preexisting sentence, evaluate the amount of "controversy" that the sentence creates, then try to improve on that score iteratively. We define a sentence as $s = (w_1, w_2, ...., w_k)$.

**Algorithm 1** Synthesize Controversial Sentences

**Require:** Sentence $s = (w_1, w_2, ...., w_k)$, Sampler function $S$, Convergence Criterion $C$

**Require:** Model List $L$, Max length $N$, How many candidates to choose $K$, Vocabulary Length $V$

$S \leftarrow \{\}$ ▷*all scores*
$m \leftarrow \max(\text{len}(s)+l, \text{N})$
**if** len(set(scores[-$C$:])) != 1 **then**
  **while** len($s$) $< m$ **do**
    $score \leftarrow G(s, L, V)$
    $i \sim \{1...len(s)\}$
    $(\tilde{w}_{1r}, \tilde{w}_{2r}, ...., \tilde{w}_{kr}) \leftarrow S(s, i, \text{True}, K)$ ▷*candidate words for replacement*
    $\tilde{\mathcal{S}} \leftarrow \{\}$ ▷*candidate sentences*
    **for** $\tilde{w}$ in $(\tilde{w}_{1r}, \tilde{w}_{2r}, ...., \tilde{w}_{kr})$ **do**
      $\tilde{s} \leftarrow s[\text{i}] = \tilde{w}$ ▷*replacement*
      $\tilde{S}$.add($\tilde{s}$)
    **end for**
    $\tilde{s} \leftarrow$ del $s$[i] ▷*deletion*
    $\tilde{S}$.add($\tilde{s}$)
    $(\tilde{w}_{1a}, \tilde{w}_{2a}, ...., \tilde{w}_{ka}) \leftarrow S(s, i, \text{False}, K)$ ▷*candidate words for addition*
    **for** $\tilde{w}$ in $(\tilde{w}_{1a}, \tilde{w}_{2a}, ...., \tilde{w}_{ka})$ **do**
      $\tilde{s} \leftarrow s$.insert($i$, word) ▷*addition*
      $\tilde{S}$.add($\tilde{s}$)
    **end for**
    $\tilde{s} \sim \{\tilde{s}_1, \tilde{s}_2...\tilde{s}_k\}$
    $score_{\tilde{s}} \leftarrow G(\tilde{s}, L, V)$ ▷*candidate score*
    **if** $score_{\tilde{s}} > score$ **then**
      $s = \tilde{s}$
    **end if**
    $\mathcal{S}$.add($score$)
  **end while**
**end if**
**return** $s$

In each iteration, a pool of candidate sentences $\tilde{S}$ is created. In order to do so, the following steps are taken. First, an index $i$ is chosen in the range of the sentence length at random. Then, the candidates are created in one of three ways: a replacement is made at $i$, an insertion is made at $i$, or a deletion is made at $i$. Then, a pool of candidate words $(\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_k)$ is sampled from the Sampler. For each word $\tilde{s}$ suggested, a new candidate sentence $\tilde{s}$ is created with either a replacement, deletion, or addition.

From the pool of candidate sentences $\tilde{S}$, a single sentence is sampled and evaluated, $\tilde{s}$. Then, the score of the sampled sentence is compared to the score of the original sentence being modified. If the modification proves to improve the divergence score, then the modification is selected, otherwise it is not. This process occurs over and over again until the convergence criterion is met, and the candidate sentence is returned

## 2.3 Sampler

In our algorithm, in each iteration we test three different options for making changes to a sentence in order to synthesize a more controversial alternative; we explore replacements, deletions, or additions. Most crucial, however, are the actual words that will be supplied for these modifications. Theoretically, this space of candidate words could be comprised of the whole vocabulary of a language. However, testing all possible words that could fit would be computationally infeasible, as well as unnecessary. Logically, certain sets of words will prove to create more or less interesting and controversial sentences. We can capture some of these sets of words with certain hypotheses. If we can propose different hypotheses that captures various subset of words, we can begin to narrow down which of these sets consistently breeds controversy. The set of words which arise from applying a hypothesis are the proposal distribution for the given space in the sentence.

Since part of our desiderata is to understand what kinds of mechanisms in language create controversy, it is important to explore multiple hypotheses for these proposal generators. For example, one hypothesis might yield words that fit well into the

current sentence but that do not increase the controversy substantially, while another finds words that clash with the rest of the sentence but cause bounds and leaps of improvement with regards to our objective function. Our goal is to be able to test out how sentence synthesis would affected by both types of proposal hypotheses. Thus, we allow the Sampler to be an argument to our platform.

The Sampler function $S$ is a stand-in for different types of proposal generators, or different ways of choosing words for candidate sentences. In this project, we have explored three different types of proposal distributions: sampling words at random, sampling from the average distribution, and sampling from BERT. As parameters, it takes in the sentence $s$, the index of the change $i$, True/False if the change being made is a replacement respectively, how many words to sample $K$.

## 2.4 Vocabulary

In order to compare the distributions produced by the models, a shared vocabulary is required. Several issues arise when trying to impose a single vocabulary. First, each model inherently has a different vocabulary. Second, each model tokenizes words differently, so a word that could actually be shared among models might look very different in two different models. Thus, the need for a shared vocabulary arises: to ensure that the distributions over words were being compared to each other accurately.

In order to do so, we predetermined a shared vocabulary, chosen from SubtlexUS, a database containing word frequencies based on English and American movies and TV series subtitles (51 million words in total). We allow the user to choose the top $V$ most frequent words. Then, we map each word of this predetermined vocabulary to how it would be tokenized by each tokenizer and get the subword tokens that compose the word. Then, we find the corresponding index of the subword tokens. In order to get back the probability of the final word, we take the sum of the log probabilities of the pieces of the words and finally normalize.

## 2.5 Decision To Accept/Reject Proposal

Once a candidate sentence $\tilde{s}$ has been produced, it must be evaluated by the Gain function $G$, yielding a score $G(\tilde{s}) = score_{\tilde{s}}$. This score must then be compared to the score of the original sentence $s$, $score_s = G(s)$. One way of deciding whether to accept or reject a sentence would be to directly compare the scores, and choose the sentence with the higher score to continue on.

However, one benefit of the Jensen-Shannon divergence was that, while it gives a sentence-wide averaged score, it also gives scores for each position of the sentence. This allows for more nuance in deciding how to accept and reject proposal sentences. Thus we can take into account how the scores vary throughout the sentence, and how a change at an index $i$ might affect the rest of the sentence after that index.

To do so, we used an exponential discounting function to get a more comprehensive score, which we ultimately directly compared, and choose the sentence with the larger score.

## 2.6 Convergence Criterion

In theory, a sentence could forever be modified. Therefore, it is essential that there is a clear stopping point, at which point a final sentence is returned. In this context, we determine a stopping point using the differences in the scores of different versions of the sentence. We allow this as a parameter that can be modified, call it $C$. Then, if the score of a sentence is the same for $C$ iterations, we return the sentence and do not attempt to modify it further. This would mean that no change made in the last $C$ iterations was able to improve the score.

## 2.7 Constraints

There were several constraints that we imposed on the algorithm otherwise in order to ensure that we were getting results that would sufficiently meet our requirements.

Our criteria are for the sentence to be minimally coherent, as well as for the score to improve substantially.

The first constraint was to impose a maximum length for the final sentence. This is because the sentence could be added to indefinitely, yielding negligibly small changes in the score at the cost of a more incoherent sentence. If there are too many additions, the sentence loses all of its original structure and becomes increasingly less coherent past a certain point. We allow this as a hyper-parameter, $N$, for the user to choose: what they think would be a reasonable maximum length for their use case.

The second constraint was imposed upon the proposal distribution, also for the sake of coherence. We restricted the proposal distribution to avoid repeated words, as well as punctuation. Added punctuation adds little to no syntactic or semantic value to the sentence, and we are specifically looking at making modifications to these sentences on the word-level, thus we chose to avoid punctuation. In addition, we filter out repeat words; in particular, BERT will occasionally suggest words that are similar to ones already in the sentence, particularly more words that are extremely popular, like "and", "the", and similar words. In order to stop sentences from having excess amounts of these over-represented words, we prevented all repeats in sentences.

# Chapter 3

# Numerical exploration

In the following section, we will show a depiction of the algorithm at work, to better understand how the algorithm iteratively creates a more controversial sentence than the starting sentence. From there, we will provide a show of convergence for the algorithm, as well as an interpretation of several of the resulting sentences, produced by different sampling functions in particular. Finally, we will summarize some qualitative results based on the kinds of sentences currently being synthesized by the algorithm. In this section, we will be showing the results obtained by comparing the following models: GPT-2, XLM, T5, Roberta, and Albert. These were chosen specifically because they represent a cohort of state-of-the-art language models, which serve as models for language processing.

## 3.1 Depiction of Algorithm

While the algorithm was laid out in Chapter 2, the best way to intuitively grasp how it works is to look through several iterations of an example sentence being modified. I will show three iterations of changes for the starting sentence "Prominent Southern Republicans were something repair a rare breed in those days". In each figure, I show the 5 words from the candidate pool for replacements, 5 words from the candidate pool for additions, then examples of how those replacements and additions would fit into the sentence. From there, the algorithm samples one such sentence and the score

is evaluated, and if the resulting score is higher than the original score, the modified sentence is accepted.

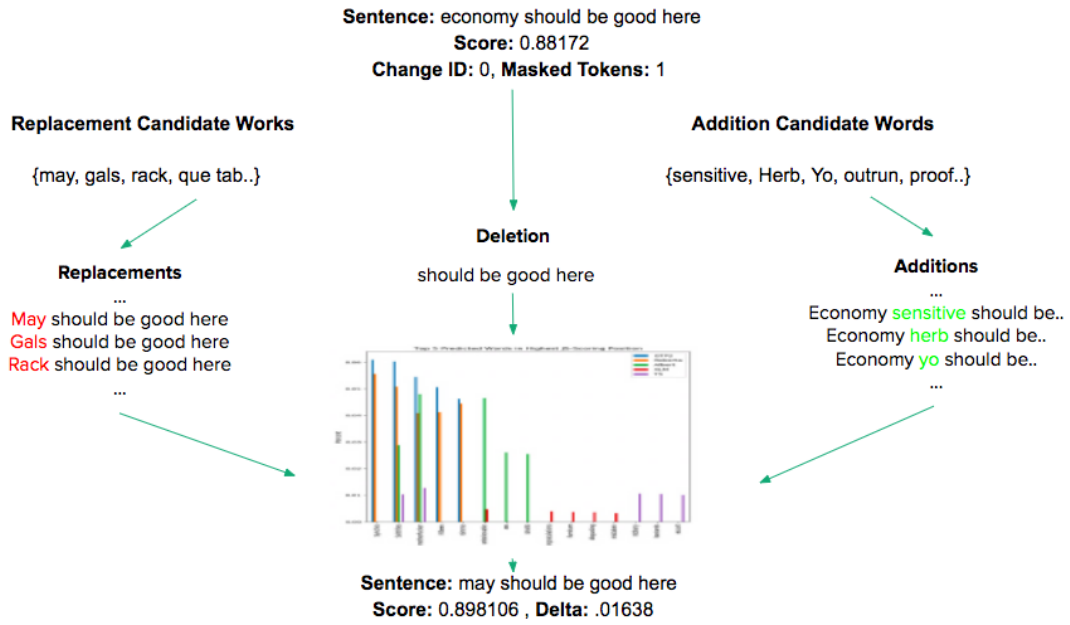


Figure 3-1: Iteration 1, with a Delta of 0.01638 in score

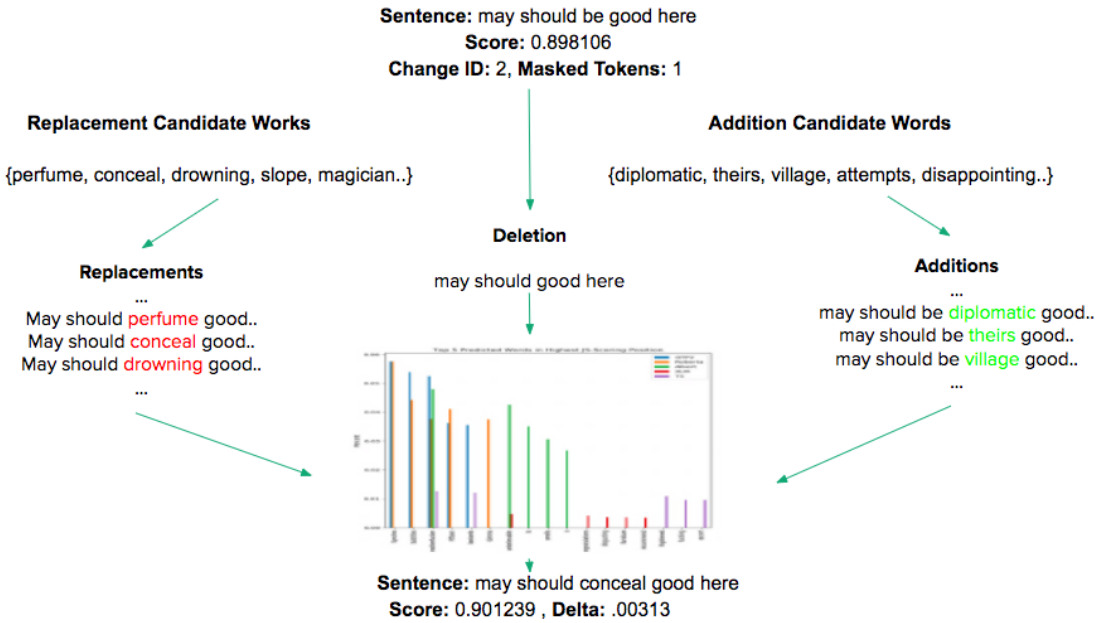


Figure 3-2: Iteration 2, with a Delta of 0.00313 in score

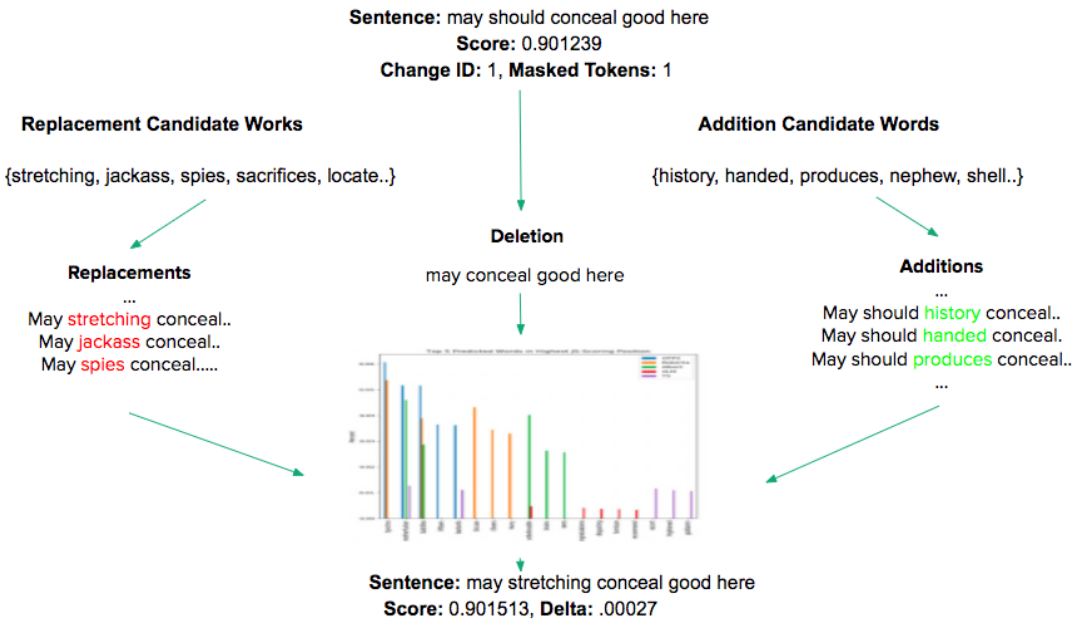


Figure 3-3: Iteration 3, with a Delta of 0.00027 in score

In order to show the controversy elicited by these modifications, I use a bar graph. This bar graph is color coordinated, where each color corresponds to a model. For the highest Jensen-Shannon Divergence scoring position in the whole sentence, this shows the top five most likely words for each model and compares words that are shared across models. Maximum controversy in this case, then, would mean that there would be 25 individual tick marks with one bar for corresponding to each word, because this would mean that the top five words found by each model were completely disjoint with the top five words of all other models. In this case, we can think of similarity as how much clustering there is (these top 5 words of course not being representative of the whole vocabulary, but to be used as an example), while difference or controversy arises in the lack of clustering. These bar graphs can be seen in greater detail in the Appendix.

## 3.2 Convergence rate

It is essential that there is a point of convergence in the scores, since otherwise a sentence could be theoretically modified indefinitely. In our platform, we allow users

to decide what convergence criterion they would like to use. This essentially means that we allow the user to decide after how many iterations of the same score do we find that the sentence has practically converged. For example, we use a convergence criterion of 1000 iterations here. In the plots below, we show an average line plot of the progression of sentence scores until the scores eventually converge. For this plot, we use the Jensen Shannon Divergence as the evaluation function, and sample from BERT for our proposal distribution.

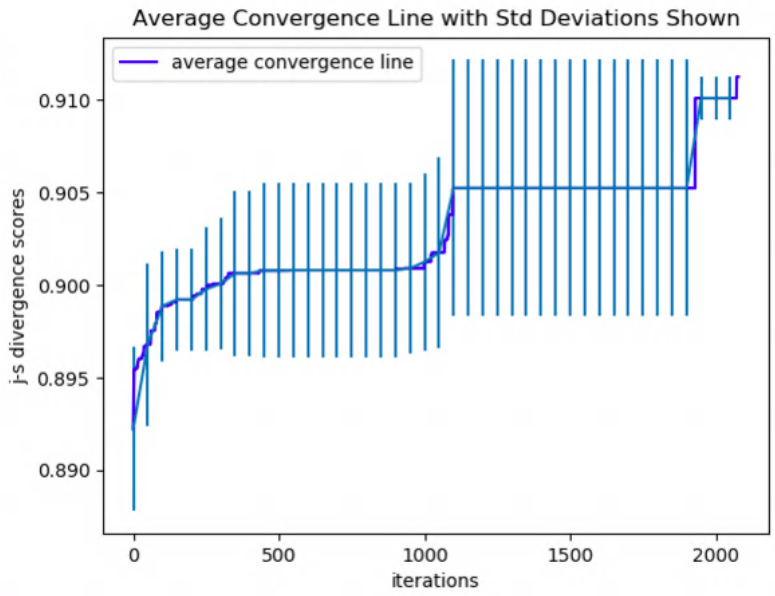


Figure 3-4: Average convergence line plot with standard deviations

## 3.3 Interpretation of resulting sentences

When looking at the final sentences synthesized by this algorithm, we must first ask ourselves: "Do these sentences meet the criteria we set out at the onset of this project?" The criteria we had imposed were that there was a substantial improvement in the score, while also retaining some level of coherence, at least to the point of looking like a "sentence" by some human-recognizable metric.

In this section, we look at some of the sentences produced, and try to draw conclusions about why they failed, so that the next iteration of sentences produced

comes closer to fulfilling our end goals.

### 3.3.1 Average Distribution Sampled Sentences

One option for a Sampler was to use the average distribution of the model list models, for the pool of candidate words. After producing several iteratoins of sentences, however, I found that the words being chosen seemed relatively arbitrary. While they did meet our desiderata of increasing controversy, they had little to no measure of coherence. Especially when making changes early on in the sentence, this would occur quite frequently, since there would be less context for the models to derive the next-word probability distributions from. For the first word in the sentence especially, this just meant that the probability distribution was based on the priors of the model rather than their language modeling abilities. Therefore, the resulting sentences were relatively uninteresting and scattered.

For example, one such sentence was: "Will racial diversity find its way to Wyoming as some people are offered jobs there ?". This eventually evolved into: "Me racial diversity pipe Chinese grad to masters grad some aff masters offered jobs there join".

### 3.3.2 BERT Sampled Sentences

In order to address this issue of context, we tried an alternative Sampler: sampling with BERT. The methodology behind this was as follows: let $i$ be the index at which a change is occurring. Replace the word at the index $w$ with a single or two masked tokens, chosen at random, the reasoning of which will be explained shortly. Pass this modified sentence to a masked language model version of BERT, and allow it to generate the top K most likely tokens that would fit in that index instead.

This approach is bidirectional, thus taking into account both the context after the index in question as well as before the index. Therefore, the predictions made were much more coherent and fit the sentences better than sampling from the average distribution, described above.

One issue that did arise was that BERT would occasionally suggest tokens that

were not equivalent to complete words. Because we are trying to modify the sentence on the word-level, we wanted the suggestions made by BERT to be words as well, instead of fragments of words. Thus, we chose the approach of choosing at random one or two masked tokens to place at the index $i$ If it was replaced by one masked token, we filter out subword-tokens (which in BERT are denoted by starting with ""). If it was replaced by two masked tokens, we ensure that the first token did not start with "", but that the second token did start with "". Occasionally, this would produce mismatched words (for example, replacing a word with "and ing"), but this happened with relatively low frequency.

Another additional difficulty that arose was that while the words suggested by the sampler fit relatively well into the sentence, often they were "uninteresting" words. For example, among the most popular words in any given situation or context are "the", "a", "and", etc. These words then become suggested quite often, and sometimes replace words with a richer content or definition.

For example, in the following sentence, "Iran" and "Russia" are meaningful words that give important context: "Therefore any attack upon Iran s nuclear facilities will bring Russia into the mess". After converging on a score, this sentence becomes the following: "and a if where political pressures on over water facilities led to two turns of fighting the an". More interesting words have been switched out, and there are more "dull" or common words in this sentence.

An additional concern which arose from this was that potentially we were inhibiting the controversy that could be realized by having sentences that fit together "too well", or "too coherently". Since BERT would always try to suggest words that could reasonably fit into the context of the sentence, it would make sense that other language models would too expect those words, and might agree on those words more often than other more varied words. Therefore, we conjectured that perhaps using BERT actually limited the accumulated controversy.

### 3.3.3 BERT with POS Sampled Sentences

In order to remedy the problem of swapping interesting or meaningful words out of a sentence, we explored an additional modification on the previous Sampler, by adding in a condition on the words proposed. For a word $w$ index $i$ let $p$ the part-of speech tag associated with $w$. Then, any words being proposed by BERT must have the same part-of-speech tag $p$ the word that it is proposed to replace. We create a dictionary of words to part-of-speech tags, using the dominant part-of-speech tag as defined by the SUBTLEXus lexicon. This is a slightly limited view, as some words change their part-of-speech tag based on the context they are being used in, but this allows us to begin to explore this modification relatively robustly.

One limitation we ran into when implementing this was that many words were not tagged properly - specifically, many words were tagged as "Unclassified". This meant that instead of being matched up to words that were similar in part of speech to the original words, they were matched to other "unclassifiable" words. In the vocabulary of the language model, this was a very varied set, which included special characters, symbols from other languages, and special tokens. Thus this approach, at the level implemented currently, led to sentences full of non-word tokens, which were not interpretable.

### 3.3.4 Random Words Sampled Sentences

In order to avoid forcing too much structure on the synthesized sentences via the candidate words proposed, as was a potential downside of sampling from BERT, we tried sampling random words. Specifically, to get a candidate set of words, we drew at random from the 10,000 most likely words in the English language, based on the SUBLEX-us corpus. This resulted in a more diverse set of sentences, and included plenty of colorful words. Though these sentences were more varied, over time they came to look more like random sets of words rather than actual sentences, as the words and structure of the original sentence disappeared through replacements, additions, and deletions since context was no longer important to choosing candidate words.

When reading these sentences, however, it does seem as though some semblance of local structure is retained.

For example, the sentence "She is also nicer when we are both alone" became "children affairs also Harry swell ahead when family are civil expression were". An additional example is "Please let me know we are eager to keep trading with Global and yourselves", which turned into "Please let me know silly are eager to returned be with value hoping assignment fingers aware settle". Some of these turns of phrases seem to have a semblance of coherence, however it is difficult to know whether this is a result of the algorithm, or whether we are noticing structure where there is none.

# Chapter 4

# Conclusions and Future Directions

In this work, we presented a novel algorithm and platform for synthesizing controversial sentences in the natural language domain. Our result differs from previous work in that it is focused on synthesizing controversial examples, and works to systematizes this process which allows us to explore the regions of language which confuse models in a scientific way. Crucially, we provide a way to test different hypotheses and methods easily and efficiently using this platform. While a concrete set of stimuli has not yet been reached, this platform has begun to give us an understanding towards what a controversial sentence should look like, what the underlying aspects of a sentence are that make it controversial, and what kinds of methods work for creating such stimuli. This understanding is necessary to knowing what the limitations and potentialities of developing such stimuli are, and knowing what sorts of methods are relevant to explore further. In this vein, there are several potential future directions to explore.

One such proposal suggests that instead of operating at the word-level, we operate at a bi- or tri-gram level. This would mean that instead of making replacements, additions, and deletions on a word by word basis, we instead use bi-grams or tri-grams instead. The reasoning behind this suggestions is that by using bi- or tri-grams, we would retain clusters of local coherence in the sentence. This may be potentially useful when testing out these sentences on human subjects; while a model does not mind if a sentence looks like a random set of words, a human may process a random set of words differently than they would process a sentence. Since these sentences

will hopefully be used to understand human language processing, we would want sentences that are processed as whole sentences to read and understand, rather than random collections of words to remember.

Another proposal includes adding a re-ranking component, once the sentences have been synthesized. This would entail trying to maximize controversy as much as possible, with little to no attention paid to retaining coherence of a sentence, in the first steps. For example, while sampling from BERT allows us to retain coherence, it might reduce controversy - indeed it likely does, since words are being purposefully chosen that could plausibly fit into the sentence, which is likely to be less controversial to the models. Instead, we may sample from a random list of words and get sentences that almost look like random lists of words, but have higher controversy scores. Then, we could evaluate these sentences post-hoc using a re-ranking algorithm, potentially employing BERT, to rank the sentences by their relative coherence. Thus, in the first step we prioritize maximing controversy, and in the second we attend to the coherence of the sentences produced.

Additional proposals include adding different types of models to the list of models being used to synthesize these sentences. Currently, we are using only large, transformer models, which could be part of why we are yielding relatively low controversy. One way to test this hypothesis is to add different types of models to the list, including RNN or LSTM-based models. One such example we are currently considering is skip-thoughts [10]. In addition, due to the flexibility of the platform, we can continue to play around with different Samplers or proposal distributions.

Finally, once we have developed examples that maximize controversy while sacrificing coherence and readability as little as possible, we will par down our set of examples, and choose a set which we will present to a cohort of human readers. As they read the sentences, we will take MRI scans of their brains. The results of Schrimpf et al [16] show that there exist models that can accurately predict neural responses, in some cases achieving near-perfect predictivity relative to the noise ceiling. Thus, we will use these results and the pipeline from this work and compare the predicted neural responses over this set of controversial stimuli, and compare it to the

MRI scans that we collect from our human subjects. In this way, we can find which models most clearly and accurately capture human brain activity elicited during language comprehension, and since the stimuli are controversial, each model's response will be substantially different from other models. This will allow us to distinctly see which model performs and predicts in the most human-like way, and hopefully reveal something about the functionality of human language processing.

# Appendix A

# Figures

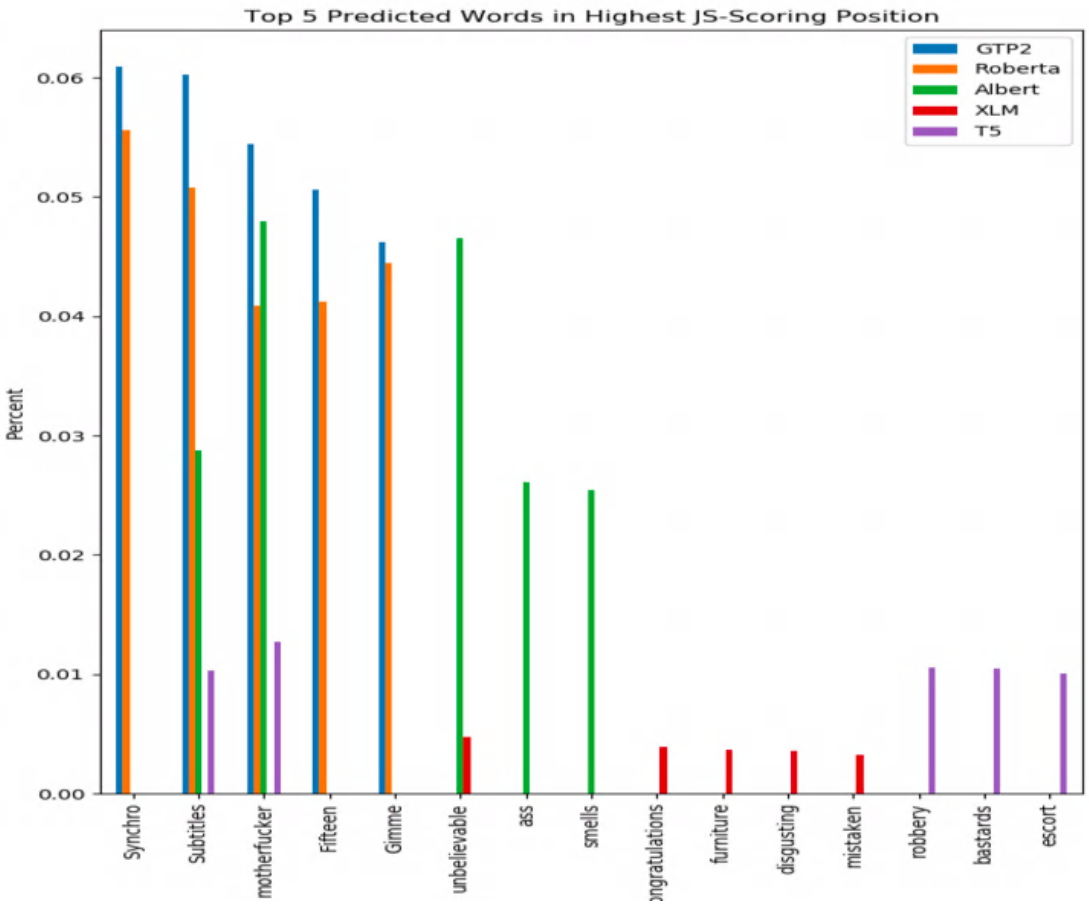


Figure A-1: Iteration 1 Controversy Graph

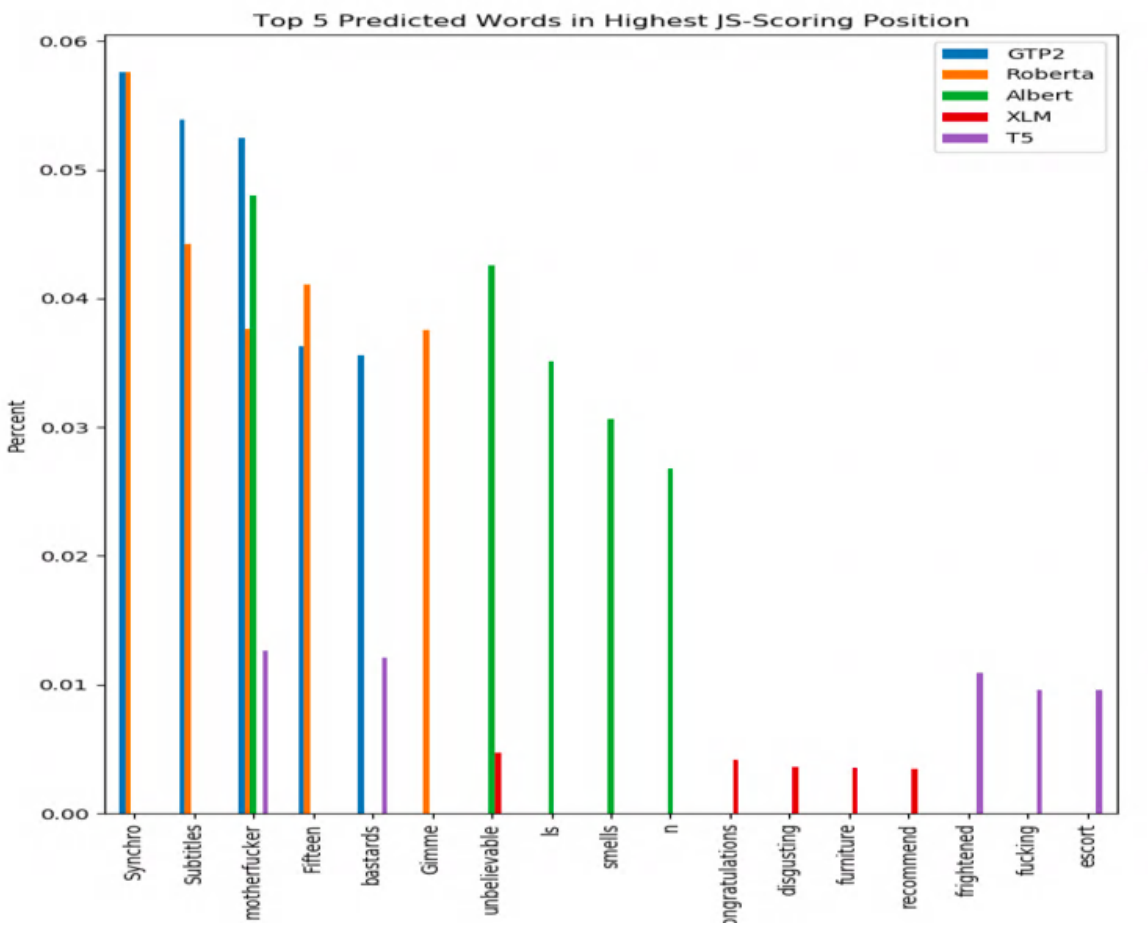


Figure A-2: Iteration 2 Controversy Graph

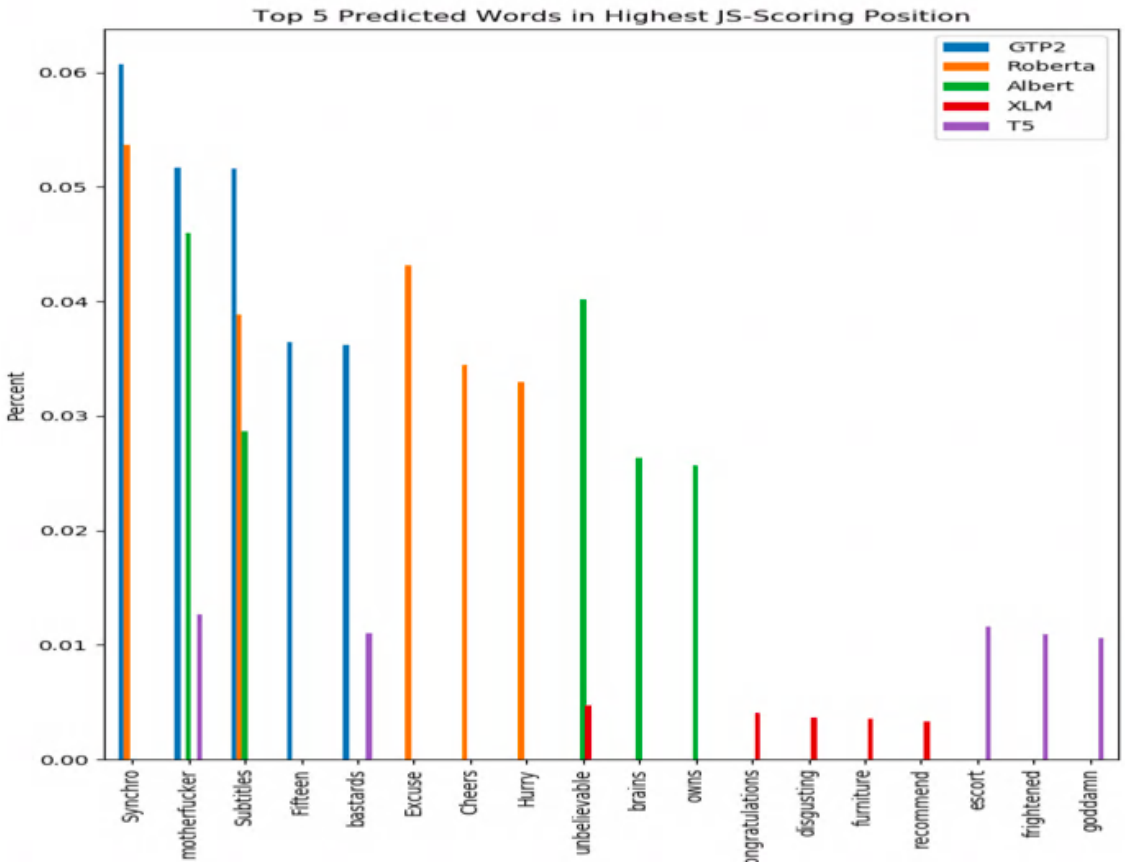


Figure A-3: Iteration 3 Controversy Graph

# Bibliography


[1] Moustafa Alzantot et al. "Generating Natural Language Adversarial Examples". In: Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2890–2896. URL: https://www.aclweb.org/anthology/D18-1316.

[2] Yonatan Belinkov and Yonatan Bisk. "Synthetic and Natural Noise Both Break Neural Machine Translation". In: *CoRR* abs/1711.02173 (2017). URL: http://arxiv.org/abs/1711.02173.

[3] Charlotte Caucheteux and Jean-Rémi King. "Language processing in brains and deep neural networks: computational convergence and its limits". In: *bioRxiv* (2020). DOI: 10.1101/2020.07.03.186288. URL: https://www.biorxiv.org/content/early/2020/07/04/2020.07.03.186288.

[4] Jenelle Feather et al. "Metamers of neural networks reveal divergence from human perceptual systems". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 10078–10089. URL: https://proceedings.neurips.cc/paper/2019/file/ac27b77292582bc293a51055bfc994ee-Paper.pdf.

[5] Evelina Fedorenko et al. "Neural correlate of the construction of sentence meaning". In: *Proceedings of the National Academy of Sciences* 113.41 (2016), E6256–E6262. URL: https://www.pnas.org/content/113/41/E6256.

[6] Tal Golan, Prashant C. Raju, and Nikolaus Kriegeskorte. "Controversial stimuli: pitting neural networks against each other as models of human recognition". In: *Proceedings of the National Academy of Sciences* (2020). URL: https://arxiv.org/abs/1911.09288.

[7] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: International Conference on Learning Representations, 2015. URL: http://arxiv.org/abs/1412.6572.

[8] Robin Jia and Percy Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017. URL: https://www.aclweb.org/anthology/D17-1215.

[9] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. "Deep supervised, but not unsupervised, models may explain IT cortical representation". In: *PLoS computational biology* 10.11 (2014), e1003915–e1003915.

[10] Ryan Kiros et al. "Skip-Thought Vectors". In: *CoRR* abs/1506.06726 (2015). URL: http://arxiv.org/abs/1506.06726.

[11] D. V. Lindley. "On a Measure of the Information Provided by an Experiment." In: *The Annals of Mathematical Statistics* 27 (1956), pp. 986–1005. URL: https://doi.org/10.1214/aoms/1177728069.

[12] Josh H. McDermott et al. "A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy". In: *Neuron* 98.3 (2018), pp. 630–644. URL: http://www.sciencedirect.com/science/article/pii/S0896627318302502.

[13] Tom M. Mitchell et al. "Predicting Human Brain Activity Associated with the Meanings of Nouns". In: *Science* 320.5880 (2008), pp. 1191–1195. URL: https://science.sciencemag.org/content/320/5880/1191.

[14] Francisco Pereira et al. "Toward a universal decoder of linguistic meaning from brain activation". In: *Nature Communications* 9 (2018). URL: https://www.nature.com/articles/s41467-018-03068-4.

[15] M R Saddler, R Gonzalez, and J H McDermott. "Deep neural network models reveal interplay of peripheral coding and stimulus statistics in pitch perception".

In: *bioRxiv* (2020), p. 2020.11.19.389999. URL: https://doi.org/10.1101/2020.11.19.389999.

[16] Martin Schrimpf et al. "Artificial Neural Networks Accurately Predict Language Processing in the Brain". In: *bioRxiv* (2020). URL: https://www.biorxiv.org/content/early/2020/06/27/2020.06.26.174482.

[17] Martin Schrimpf et al. "Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like?" In: *bioRxiv* (2018). URL: https://www.biorxiv.org/content/early/2018/09/05/407007.

[18] B. Smucker, M. Krzywinski, and N Altman. "Optimal experimental design". In: *Nature Methods* 15 (2018), pp. 559–560. URL: https://doi.org/10.1038/s41592-018-0083-2.

[19] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014. URL: http://arxiv.org/abs/1312.6199.

[20] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762.

[21] Daniel L. Yamins et al. "Hierarchical Modular Optimization of Convolutional Networks Achieves Representations Similar to Macaque IT and Human Ventral Stream". In: *Advances in Neural Information Processing Systems*. Vol. 26. 2013, pp. 3093–3101. URL: https://proceedings.neurips.cc/paper/2013/file/9a1756fd0c741126d7bbd4b692ccbd91-Paper.pdf.

[22] Daniel L. Yamins et al. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8619–8624.

[23] Huangzhao Zhang et al. "Generating Fluent Adversarial Examples for Natural Languages". In: Association for Computational Linguistics, 2019, pp. 5564–5569. URL: https://www.aclweb.org/anthology/P19-1559.

[24] Zhengli Zhao, Dheeru Dua, and Sameer Singh. "Generating Natural Adversarial Examples". In: *CoRR* abs/1710.11342 (2017). URL: http://arxiv.org/abs/1710.11342.

[25] Chengxu Zhuang et al. "Toward Goal-Driven Neural Network Models for the Rodent Whisker-Trigeminal System". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017, pp. 2555–2565. URL: https://proceedings.neurips.cc/paper/2017/file/ab541d874c7bc19ab77642849e02b89f-Paper.pdf.