The Hilarious Bluffing GAN

Stanford CS224N Custom Project

Isaac Supeene
Department of Computer Science
Stanford University
isupeene@stanford.edu

Abstract

In this work, I apply NLP to the world of AI gaming by developing an AI for the classic word game Balderdash. The game can be decomposed into two tasks, both conditioned on an obscure prompt word: a) generate a convincing fake definition for the word, and b) distinguish real from fake definitions. My method, using a naive one-hot char encoder paired with a BART decoder, and trained with a novel Contrastive GAN, achieves human-level performance.

1 Key Information to include

- Mentor: Manan Rai
- External Collaborators: None
- Sharing project: No

2 Introduction

AI methods are frequently employed to play games ranging from Chess and Go to Starcraft. These techniques typically involve reinforcement learning and tree searches. Some even include computer vision technology, for example those that learn to play Atari games given the on-screen pixels as input. To the best of my knowledge however, natural language processing has not yet been incorporated into AI gaming tasks.

In this work, I have created an AI capable of matching average human performance in the classic word game Balderdash, in which players generate fake definitions for words, and try to distinguish real from fake definitions. This is an interesting problem, since it tests the AI's ability to understand subword-structures, which is key for producing a convincing definition. I demonstrate the AI's ability to learn such subword features even from a very primitive encoding scheme, and explore a novel Contrastive GAN for co-training the generator and discriminator.

2.1 The Game

In Balderdash, each round one player (the 'dasher') draws a card with 5 obscure words & their definitions, chooses one of them as the prompt, and reads it aloud (along with its spelling) to the other players. Each other player submits a fake definition to the dasher, who then reads out all the fake definitions and the real definition in random order. Players then try to guess which definition is real. The goal is to fool your opponents, while guessing the right answer yourself.

My implementation has a couple of minor differences. For one thing, no player needs to be the dasher each round, as this task can be automated. Also, as noted in the Data section, training and evaluation took place not with the 'official' Balderdash dictionary, but with a publicly available list of roughly 17,000 obscure words.

This translates into two distinct tasks for the AI:

1. Given a prompt word, generate a convincing fake definition.
2. Given a prompt word and a sequence of definitions, identify the real definition.

The latter task can be reduced to a binary classification task of each definition as real or fake. In gameplay, the definition with the highest likelihood of belonging to the 'real' class is chosen as the AI's guess.

3 Related Work

The most closely related existing work I could find was thisworddoesnotexist.com. Although its main feature is the ability to generate and define fake words, it also has the ability to generate definitions for words entered by the user. Some of its definitions are quite convincing, and others are less so. It does appear to exhibit good responsiveness to common word parts, such as -phobia, -ology, and -ocracy. A small sample of generated definitions are presented in Appendix B. In this sample, we can observe that my Bluffing GAN and thisworddoesnotexist.com make similar use of subword structure, but that thisworddoesnotexist.com sometimes produces incoherent results. The Bluffing GAN on the other hand, has a bad habit of using memorized definitions from the training data.

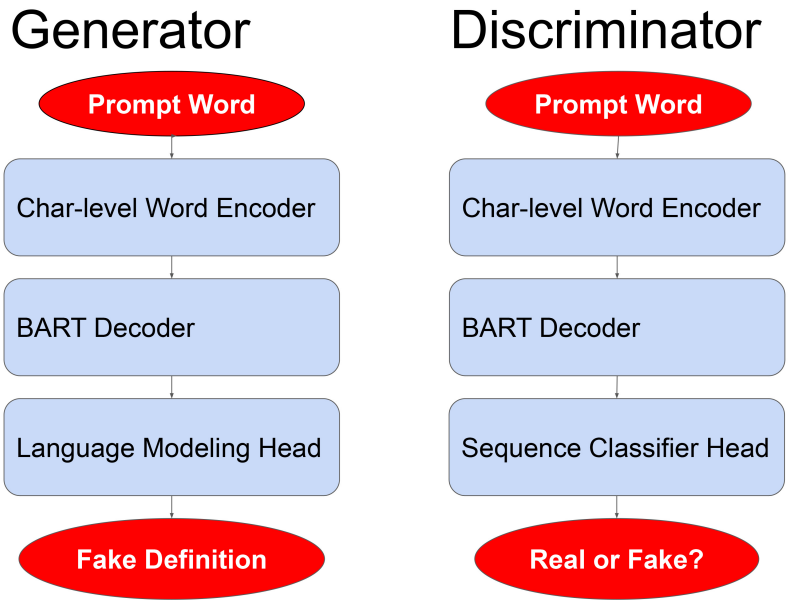
The state of the art for character-level encodings is Banar et al.'s CharTransformer [1]. My char encoder is quite primitive in comparison, so there is clearly headroom to improve the model.

A related approach to using GANs for text generation was proposed by Subramanian et al. [2]. Their approach is to have the discriminator operate on the entire softmax output of the generator, which allows the discriminator score to be propagated all the way back, as in a conventional GAN. A notable limitation of this approach is that subsequent tokens in the generation process are still conditioned on previous tokens, where as in my Contrastive GAN, a number of totally independent samples are produced and scored separately.

4 Approach

The game consists of two sub-tasks: generation of a fake definition, and discrimination between real and fake definitions. Therefore, I trained two separate models, a generator and a discriminator.

Figure 1: Bluffing GAN architecture



The baseline models are based on GPT-2, with the generator using a language-modeling head, and the discriminator using a sequence classification head. All the weights of the generator, including the

LM head, were loaded pretrained using the HuggingFace transformers library, and the LM head was finetuned on true definitions. The discriminator used all pre-trained weights except for the sequence classification head, which was trained from scratch.

There are two incremental improvements that were added to the baseline: The GAN training, and the char-encoder. This results in four total variations:

1. Unconditioned, no GAN (the 'Baseline' model)
2. Unconditioned, trained via GAN (the 'GAN' model)
3. Conditioned on the prompt word via the char-encoder (the 'Conditioned' model)
4. Conditioned on the prompt word via the char-encoder, and trained via GAN (the 'Bluffing GAN')

4.1 The Contrastive GAN

I introduce a novel contrastive loss function for adversarial text generation. Although it demonstrates a nominal increase in model quality, there are still unresolved training instability issues that lead to text degeneracy when training for multiple cycles.

The driving intuition is that by correlating its likelihood scores with the discriminator scores, the generator is more likely to produce text that fools the discriminator. Concretely, for any given sequence, let G_i be the generator's likelihood of producing the ith token, given the context of the previous tokens. The generator score for the sequence, S_G is then $\sum_{i=0}^{length} log(G_i)$, that is, the log probability of the sequence. Meanwhile, the discriminator score S_D is the probability assigned to the 'true' class.

These scores are produced for a batch of N sequences, and the loss is defined by:

$$L = \sum_{j=0}^{N} abs\left(\frac{S_{G_j} - min(S_G)}{max(S_G) - min(S_G)} - \frac{S_{D_j} - min(S_D)}{max(S_D) - min(S_D)}\right)$$

In other words, the loss is the sum of the absolute differences between the normalized generator scores and normalized discriminator scores.

	Normalized Generator Score	**Normalized Discriminator Score**	**Loss**
Definition 1	1.0	0.5	0.5
Definition 2	0.7	0.8	0.1
...			
Definition N	0.0	0.7	0.7

Table 1: Illustration of the Contrastive Loss Function for Text Generation

At each discriminator epoch, we gradually replace fake definitions in the discriminator's training set with new definitions produced by the latest generator. The fraction of the false examples that are replaced is a hyperparameter called the 'replacement rate'. Training alternates between one epoch of generator training and one epoch of discriminator training. The generator training phase is controlled by two additional hyperparameters - number of batches per epoch, and number of definitions generated per batch. Each batch is a set of definitions all associated with a single word, and the loss function is applied for the whole batch. Hyperparameter values used in training are described in Section 5.

Since the total loss increases with the batch size, you could imagine that the optimal learning rate is strongly affected by the batch size, or that the total loss should be normalized by the batch size in some way. I have not had the opportunity to test these approaches.

There is an important implementation detail to note regarding the contrastive loss function. The sequence loss returned by the HuggingFace transformers library is not calculated in a differentiable way due to a) in-place operations, and b) a torch.no_grad decorator on the generate function. I needed to hack my local transformers installation to resolve these issues, so the accompanying code will not work out-of-the-box.

4.2 Prompt-conditioned Models

My approach to conditional text generation was to pair a BART decoder with a custom character-level encoder. This contrasts with the approach of thisworddoesnotexist.com, which uses a decoder-only approach to conditional generation, by tokenizing the prompt word and directly using the tokenized IDs as input to the decoder.

The character encoder is a simple one-hot encoder. BART's hidden states are 1024 elements long, but my input space only includes 26 lower-case letters and the hyphen, so I only use the first 27 elements of the encoder, plus an extra element to use as a pad token. In order to learn a usable signal from this simple encoding, I finetuned the cross-attention weights of the decoder along with the language-modeling head. As I will show below, even this very simple approach produced a remarkable ability to understand the substructure of words, and generate definitions accordingly.

One additional obstacle to this approach is the fact that BART does not come with a pretrained language-modeling head. When attempting to use the BART decoder to generate definitions in an unconditioned way, I observed substantial regressions from the GPT-2 Baseline. However, by pre-training the LM head on one 1200th of Wikipedia for 20 epochs, I was able to close the gap to the baseline. This pretrained model was used as the base for subsequent finetuning on the Unusual Words dataset.

The default data collator from HuggingFace skips string-valued model inputs. To get the words into the model, I wrote my own data collator which is essentially a copy-paste of their default_data_collator with string-skipping logic removed.

The modeling code used for the word-conditioned model is highly derivative of HuggingFace code copied from modeling_bart.py in the transformers library.

5 Experiments

5.1 Data

Real examples of word-definition pairs are drawn from the Dictionary of Unusual Words [3]. Although not all of these words rise to the same standards of obscurity as those in the boxed Balderdash game, most of the words were new to me, and very few of these new words have entirely obvious definitions based on their structure.

The following preprocessing was done to the data:

1. All words containing any characters other than [a-z], [A-Z], or '-' were dropped from the corpus. This includes such characters as 'â', 'é', or 'ñ', as well as the space character. Since there were only 8 space characters in all 17003 words, I judged that this is too rare for the encoder to learn anything useful about them.
2. All words are rewritten in entirely lowercase (meaning the total character set of the vocabulary is reduced to [a-z] and the hyphen).
3. Words with fewer than 3 or greater than 20 characters were removed. There were 8 2-letter words, and a total of 11 words with more than 20 characters.
4. All words with definitions containing '^', '•', or '…' were excluded. These were very rare, and their usage was atypical. For example, '^' was used in the definition of 'lamboid' to describe what the Greek letter lambda looks like.
5. '“' and '”' were replaced with '"' in all definitions. (In case this is not clear, 'opening' and 'closing' double-quotes are replaced with generic straight double-quotes.)
6. Definitions containing ';' were split into separate examples. All examples of definitions for the same word are placed into the same dataset (train, dev, or test). In an ideal world, I would like to have kept definitions together if they are rephrasings of the same word sense (abditive: remote; secretive; hidden) and separate them if they are separate word senses (accolade: curved architectural moulding; vertical line joining two musical staves). However, I did not have the time to perform this task, as there are over 3500 semicolons in the dataset.

The total list of exclusions is given in Appendix D. The final dataset size after exclusions and splitting definitions on semicolons is 20,391. These were divided randomly into train (14,258 definitions), dev (2100 definitions), and test (4033 definitions).

Additionally, besides the test set which is drawn from the same distribution as train and dev, there is a special 'Grande Reserve' dataset: the Dictionary of Lost Words [4], which is also from the Phrontistery. This dataset was used to make the final evaluation against the 'average human', and I feel like it represents the distribution of the boxed Balderdash game better than the main dataset. This dataset contains 266 words, and was manually curated to exclude words which are too obvious (boreism: behaviour of a boring person) or inappropriate for a public demo (surgation: erection of the penis).

In addition to true examples, the discriminator requires fake examples to train on. These were created using two early baseline generators. The first (V0.0) was trained with a tokenization bug, and produced notably poor definitions. The second (V0.1) was identical to the Baseline, except that all weights were finetuned, rather than only the language-modeling head. This results in reasonable, but still inferior definitions. In the case of training the Conditioned discriminator, fake definitions were paired with a random word from the training set.

Generator V0.0	**Generator V0.1**
There is an image	orgy of laughter
"But there was one thing	The most powerful word in a particular sentence or phrase
What is food	somewhat deep bass

Table 2: Generated fake examples used for train & eval of the Baseline discriminator

5.2 Evaluation method

There is no inherent 'ground truth' for either the generator task or the discriminator task. Although there are true definitions, the generator succeeds by fooling its opponent, not by producing the true definition. Furthermore, there is no existing corpus of fake definitions against which the discriminator can be evaluated; its evaluation is necessarily relative to the source of negative examples.

I have defined two metrics for evaluating the AI. The 'accuracy' metric evaluates the discriminator, and the 'persuasiveness' metric evaluates the generator. When evaluated in a pairwise fashion, a generator's persuasiveness is the complement of a discriminator's accuracy. To compare different discriminators, we must evaluate them sequentially against the same generator, and to compare different generators, we must evaluate them against the same discriminator.

Alternatively, we can evaluate one generator-discriminator pair against another. This is analogous to two humans playing against each other, and this mode of evaluation is the one upon which my claim of 'human-level' performance is based. In this case, each generator-discriminator pair gets an accuracy score (which is the complement of the other pair's persuasiveness score), and a persuasiveness score (which is the complement of the other pair's accuracy score). Because of the relations between the scores, one pair will always have both greater accuracy and greater persuasiveness, and this pair is the superior player overall.

There is an additional distinction between automatic and manual evaluation. In the case of automatic evaluation, one discriminator is paired against one generator, and we produce accuracy and persuasiveness metrics for that pair of models. In the case of manual evaluation, a human plays against a generator-discriminator pair, and persuasiveness and accuracy of the AI can be determined relative to the human opponent.

There is a final metric that I have created to probe the discriminator: subword sensitivity. I created a small evaluation set of 35 words with clear hints as to their definitions in the word. Each word is associated with two definitions. Neither is the true definition, but one is the definition of a word with the same meaningful subword, and the other is an unrelated definition. This dataset is presented in Appendix C. Subword Sensitivity is the proportion of examples where the discriminator thinks the relevant definition is more probable than the irrelevant one.

5.3 Experimental details

I have evaluated 4 different models against 2 different opponents. The models are as follows:

1. Baseline: Model trained with no prompt conditioning, and no GAN.
2. GAN: Model trained via the contrastive GAN, but with no prompt-conditioning.
3. Conditioned: Model conditioned on the prompt word, via the one-hot char encoder.
4. Bluffing GAN: Model conditioned on the prompt word and trained via GAN.

The two opponents are the Baseline model and the Human Expert (i.e. the author). Additionally, the Bluffing GAN was evaluated against CS224N participants via an interactive demo at the poster session.

Each model is evaluated against the Baseline for 100 rounds, and against the Human Expert for 25 rounds. The interactive demo comprised 20 rounds, all of which are detailed in Appendix A.

5.3.1 Training Regimes

All experiments were performed with the default learning rate from the transformers library of 5×10^{-5}. Training was done on a GTX 3090.

The training regime for the Baseline generator was a language-modeling task with all weights except the LM head frozen. The examples were real definitions from the Unusual Words dataset. This ran for 40 epochs. The Baseline discriminator was trained to classify real and fake examples, with the fake examples produced using the method described in Subsection 5.1. It was also trained for 40 epochs, with only the sequence classification weights being updated. Each of these took about half an hour to train.

The GAN model was trained using the Baseline model as the starting checkpoint. As I had not yet developed the replacement rate technique, the effective replacement rate was 100% - that is, the entire set of fake examples was replaced by examples from the generator. Each generator training epoch consisted of 100 batches with 15 examples per batch. The model improved after one cycle of training, then began to degenerate. Therefore, the model used for evaluation is the result of 1 full cycle of training both the generator and the discriminator. This took about 20 minutes to train.

The Conditioned model required an additional pre-training step due to the lack of a pretrained LM head for BART. My initial results showed a large qualitative drop in definition generation quality when comparing an unconditioned BART decoder to the Baseline generator. I resolved this issue by pre-training the BART decoder's LM head on a language-modeling task on one 1200th of wikipedia for 20 epochs. After doing so, I fine-tuned the model on the definition generation task and observed qualitatively similar results to the Baseline. The pretraining took approximately 6 hours.

After closing the gap between BART and the baseline, the next step was to add the word encoding. My approach was to use a one-hot character embedding vector to represent each character, and fine-tune the cross-attention weights and language-modeling head. I trained the generator for 100 epochs, which took about 1.5 hours. The discriminator was similarly trained with the cross-attention weights and sequence classification head being updated. The discriminator was trained for 20 epochs, which took about 20 minutes.

Finally, the Conditioned model was used as the base for training the Bluffing GAN. In this experiment, I was able to run the GAN for 3 cycles without degeneration by freezing the language-modeling & sequence-classification heads and only training the cross-attention weights. I did not have time to attempt to train the model for longer, or further tune the hyperparameters. The replacement rate was 5%, and each generator training epoch consisted of 100 batches with 15 examples per batch. Training took about 1.5 hours.

5.4 Results

I report the results of evaluating all models against the baseline and the Human Expert, and the Bluffing GAN's performance against a collection of CS224N participants.

First, the positive and expected findings: the Bluffing GAN exhibits the highest persuasiveness of any model when playing against the Human Expert. Further, it achieved an accuracy of 55% and

	Baseline	Human Expert	CS224N Participants
Baseline	Accuracy: 96% Persuasiveness: 3%	Accuracy: 56% Persuasiveness: 0%	N/A
GAN	Accuracy: 80% Persuasiveness: 9%	Accuracy: 52% Persuasiveness: 0%	N/A
Conditioned	Accuracy: 62% Persuasiveness: 53%	Accuracy: 44% Persuasiveness: 4%	N/A
Bluffing GAN	Accuracy: 66% Persuasiveness: 52%	Accuracy: 40% Persuasiveness: 16%	Accuracy: 55% Persuasiveness: 40%

Table 3: Final evaluation results. The leftmost column indicates the model under evaluation, while the topmost row indicates the evaluation adversary. Evaluation against the Baseline and the Human Expert used the Test split. Evaluation against CS224N Participants used the Lost Words dataset.

	Baseline	GAN	Conditioned	Bluffing GAN
Subword Sensitivity	43%	49%	40%	60%

Table 4: Subword Sensitivity. The topmost row indicates the model under evaluation. The sensitivity score is the percentage of time the model preferred a related definition to an unrelated definition.

a persuasiveness of 40% out of 20 rounds against CS224N participants. This means that the AI was short of human performance by a single point. Although the humans did outperform the AI in aggregate, 7 of the 20 rounds were played by course staff, including 3 from Chris Manning, none of whom (by my recollection) dropped a single round to the AI. Under these conditions, I think these results justify the claim that average human-level Balderdash performance has been attained, though it has a long way to go before reaching the level of human experts.

Additionally, the Bluffing GAN achieves the highest subword sensitivity, at 60%, further demonstrating the promise of the Constrastive GAN method.

Now for the negative, unexpected, or unexplained results: for unknown reasons, model accuracy actually drops as the sophistication of the model increases. This is totally counter to expectations, and suggests that my modeling decisions may have been too strongly based on the subjective quality of the generated text, and not enough on the power of the discriminator. This pattern appears against both the Baseline and the Human Expert.

Subword sensitivity is lower than expected, even for the Bluffing GAN, at 60%, and are shockingly low for the Conditioned model at 40%. Perhaps this is somehow related to the poor accuracy stats described above. Obviously, Baseline and GAN models are not conditioned on the prompt word, so their scores are entirely due to chance. Since the results vary so much, its likely that a larger, better-curated test set for subword sensitivity is needed to draw any real conclusions.

6 Analysis

On the Generator side, my main observation is how often the model produces definitions from the training data. Approximately 95% of the definitions generated by the Bluffing GAN are true definitions from the training set, applied to newly seen words. A little more creativity would be desirable. This applies to both the Conditioned model and the Bluffing GAN, with the Bluffing GAN seeming to draw upon more relevant definitions, resulting in its higher persuasiveness. This must be the result of the additional GAN training, in which the cross-attention weights were fine-tuned to try to fool the discriminator.

Considering the crudity of the encoder, the generator is notably quite sensitive to word parts. Appendix B shows some example definitions that were provided to the Bluffing GAN and to thisworddoesnotexist.com, and the Bluffing GAN's ability to be informed by the nature of the prompt word is remarkable. It is therefore all the more disappointing to see that the discriminator doesn't exhibit the same sensitivity.

The discriminator did not reach a level of quality that I'm satisfied with. I was particularly disappointed in the subword sensitivity score. In this test, I found that the Bluffing GAN was excellent with

-phobia, -ocracy, and -iferous words, but no better than chance at various other word parts like -meter, -icide, -scope, and -form. It is unclear what issues with the model leads to these results, and my lack of insight is due to focusing almost entirely on the generation portion of the task, probably just because it's more interesting to interact with the generator. With careful attention to the discriminator, I think its performance could be improved dramatically.

7 Conclusion

I have presented a simple Balderdash AI using a one-hot char encoder and a BART decoder, and introduced Contrastive GAN for text generation. Although the techniques employed by my model fell short of my ambitions, the results when evaluated against CS224N participants are very exciting, and indicate that the AI is performing at a level of skill approximating that of the average human.

One notable success is the demonstration that a one-hot char encoder is sufficient for the BART decoder to learn a substantial amount about the structure of subwords. This can be seen especially well in Appendix B, in which the generated text bears a strong relation to the word form. Surprisingly, this was not reflected as well in the discriminator's subword sensitivity score, which suggests that there is a lot of headroom for improving the discriminator.

Although I introduced the Contrastive GAN, I was not able to develop or prove the effectiveness of this approach to my satisfaction. Nonetheless, I believe it is a promising approach, as it did result in a substantial increase in both subword sensitivity, and persuasiveness against the Human Expert, as noted in Section 5.4.

Aside from issues with the GAN, other weaknesses of the project include the ad hoc and subjective metrics, the primitive one-hot encoding scheme, and the tendency of the generator to produce memorized definitions from the training set.

References

[1] Nikolay Banar, Walter Daelemans, and Mike Kestemont. Character-level transformer-based neural machine translation. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, NLPIR 2020, page 149–156, New York, NY, USA, 2020. Association for Computing Machinery.

[2] Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron Courville. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 241–251, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] Dictionary of unusual words. https://phrontistery.info/ihlstart.html, 2021.

[4] Compendium of lost words. https://phrontistery.info/clw.html, 2021.

A Interactive Demo

Word	True Definition	Player Definition	AI Definition	Player Correct	AI Correct
soleated	shod like a horse	containing sunlight or sunshine	to befool	Yes	Yes
aeipathy	an unyielding disease	having a dislike for people	loss of strength	Yes	Yes
gelicide	a frost	killing of insects	killer or destroyer of laws	No	Yes
secability	capability of being cut	the ability to seek new information	condition of having abnormally large digits	No	No
jecorary	of or relating to the liver	being elusive	hackney-coach	No	Yes
stigmatypy	printing portaits using dots of different sizes	being prejudiced against handwriting	drop by drop	Yes	Yes
venalitious	of the sale of humans as slaves	pertaining to physical substance	indicating juxtaposition	Yes	Yes
vultuous	having a sad or solemn countenance	having the characteristics of a vulture	dung-eating	Yes	No
antipelargy	reciprocal or mutual kindness	opposition to rule by farmers	the act or study of kissing	Yes	Yes
lignatile	living or growing on wood	having the properties of a volcanic rock	person of the same age	Yes	Yes
phalerate	ornamented	to aerate by means of shaking and forced air	treatment by mud baths	Yes	No
rupography	art of taking impressions of coins or medals in sealing wax	the study of puzzle generation	effect of physical emanations on photographic plates	Yes	No
apanthropinization	withdrawal from human concerns or the human world	the process of unifying geographical regions	craze for writing	Yes	No
novaturient	desiring changes or alterations	novel nutrient	reluctant	No	No
rendling	curdling or setting of cheese	slapping someone on the face	a very little kid	No	No
gutterniform	shaped like a water pitcher	poorly dressed officer	shrubby	No	Yes
lagenarious	flagon-shaped	consisting of or related to the field of malt beverages	bearing a whip	Yes	No
latibule	hiding place	hot dog	to provide with a tube	No	No
phlyarologist	one who talks nonsense	people	study of the sense of smell	No	Yes
vocitate	to name or call	have an audacious argument	carrying or leading	Yes	Yes

Table 5: Record of interactive demo, which provides a head-to-head comparison between the AI and human players.

B Comparison with thisworddoesnotexist.com

Word	True Definition	Bluffing GAN	thisworddoesnotexist.com
gerascophobia	fear of growing old	fear of work	strong, irrational fear, especially of being sexually promiscuous or promiscuous
halomancy	divination using salt	divination by means of a fingernail	the tendency of a person to avoid physical contact with their partner
neossology	study of nestling birds	study of reproduction and heredity	the branch of zoology concerned with the microscopic description of living tissues
mesocracy	government by the middle classes	conclusion or corollary	another word for monarchy, usually having a form representing a central ruler
clinophilia	passion for beds	love or fondness for dogs	sexual interest in or fondness for clothes
fistuliform	shaped like a pipe	shaped like a bristle	relating to or characteristic of a fist
libaniferous	yielding or bearing incense	of, like or pertaining to thunder and lightning	bearing only part of the body
nephalism	total abstinence from alcoholic drinks	belief in the existence or importance of spiritual entities	a tendency to treat (someone or something considered to be wrong) with dislike or guilt
novaturient	desiring changes or alterations	having four wings, such as a moth or butterfly	of or denoting a form of natural light in which it does not pervade or mirror the earth's surface, but does appear to shine outward

Table 6: Comparison between Bluffing GAN and thisworddoesnotexist.com. Note that all the Bluffing GAN's definitions except for 'halomancy' are memorized definitions from the training data.

C Subword Sensitivity Dataset

Word	Relevant Definition	Irrelevant Definition
acrophobia	fear of crossing busy streets	premium paid on foreign currency exchange
ailurophobia	fear of walking	moulding diverse ideas into one
erotophobia	fear of computers	due to external forces or causes
clinophilia	any abnormal sexual attraction	study of unexplained mental phenomena
anthropobiology	study of anaesthetics	name written backward
cetology	science of the geographic description of anything	shaped like a funnel
codicology	study or theory of the basis of knowledge	killing of bishops
areometer	instrument for measuring radiant energy or infrared light	disgrace
coulombmeter	instrument for measuring osmosis into a solution	magic lantern for projection
galactometer	instrument for measuring electrical current	store of anything
exophagy	practice of feeding on soil	plaster of Paris used in painting
chromatocracy	rule by beasts	well-read individual
argentocracy	government by none	fear of heights
technocracy	government by strumpets	bearing or having runners
substantialism	belief in indifference to pleasure or pain	to subdue
secularism	doctrine that objects of cognition are real	becoming rancid
quietism	belief in universal soul	hallucination
salutiferous	bearing petals	to speak grandiosely or grandiloquently
papuliferous	bearing ozone	divination using the heavens
nuciferous	yielding or bearing incense	a mistress
muricide	killing of larvae	cross-country skiing or running
formicide	killing of people because of their political beliefs	many-stringed instrument like a lute
siblicide	killing or killer of a bear	acquisition of property by long usage and enjoyment
vortoscope	instrument for detecting earthquakes	long discussion
scotoscope	instrument for viewing the interior of the eye	ability to satisfy
nephoscope	instrument for viewing interior of peritoneal cavity	unit of brightness of light
lachrymiform	shaped like a plate or layer	wool-bearing
ovopyriform	nipple-shaped	tentative
penniform	shaped like a long nose	quilted armour with studs
balneography	art of printing in colour using wood	unit of length equal to 22 yards
chalypsography	writing or written work describing chronological events	eating dirt
dittography	art of engraving on gypsum	triangular heraldic charge
hydrotaxis	loving or preferring water	sandstone material used to scrape ships' decks
iconology	a taste for pictures and symbols	the worship of fish
anthropophagous	knowledge of the nature of humanity	doctrine of the rejection of moral law

Table 7: Bespoke dataset compiled to test the discriminator's subword sensitivity.

D Exclusions

accollé: Illegal character in word
acharné: Illegal character in word

affronté: Illegal character in word
ai: Too short
aiué: Illegal character in word
an: Too short
animé: Illegal character in word
antidisestablishmentarianism: Too long
appaumé: Illegal character in word
appointé: Illegal character in word
bêtise: Illegal character in word
bienséance: Illegal character in word
bombé: Illegal character in word
bonbonnière: Illegal character in word
borné: Illegal character in word
botonée: Illegal character in word
bouclé: Illegal character in word
bourrée: Illegal character in word
broché: Illegal character in word
cabré: Illegal character in word
camaïeu: Illegal character in word
chambré: Illegal character in word
chômage: Illegal character in word
coulée: Illegal character in word
dancetté: Illegal character in word
declassé: Illegal character in word
dégagé: Illegal character in word
dégringolade: Illegal character in word
démarche: Illegal character in word
démenti: Illegal character in word
depaysé: Illegal character in word
désobligeante: Illegal character in word
détraqué: Illegal character in word
diamanté: Illegal character in word
donné: Illegal character in word
éboulement: Illegal character in word
ébrillade: Illegal character in word
éclaircissement: Illegal character in word
éclat: Illegal character in word
écorché: Illegal character in word
écrevisse: Illegal character in word
écuelle: Illegal character in word
eellogofusciouhipoppokunurious: Too long
élan: Illegal character in word
electroencephalograph: Too long
ellipsis: Illegal character in definition
éloge: Illegal character in word
em: Too short
émeute: Illegal character in word
éolienne: Illegal character in word
éprouvette: Illegal character in word
espiègle: Illegal character in word
étui: Illegal character in word
évolué: Illegal character in word
fainéant: Illegal character in word
ferronière: Illegal character in word
flânerie: Illegal character in word
flèche: Illegal character in word
floccinaucinihilipilification: Too long
foulé: Illegal character in word
galère: Illegal character in word

garçonniere: Illegal character in word
genouillère: Illegal character in word
guéridon: Illegal character in word
guérite: Illegal character in word
gynotikolobomassophile: Too long
haèek: Illegal character in word
hérissé: Illegal character in word
heterotransplantation: Too long
honorificabilitudinity: Too long
hysteron proteron: Illegal character in word
interpunct: Illegal character in definition
jardinière: Illegal character in word
lambdoid: Illegal character in definition
lamé: Illegal character in word
lindy hop: Illegal character in word
malgré: Illegal character in word
manège: Illegal character in word
manqué: Illegal character in word
matelassé: Illegal character in word
mésalliance: Illegal character in word
métayage: Illegal character in word
métier: Illegal character in word
moiré: Illegal character in word
mouillé: Illegal character in word
nécessaire: Illegal character in word
négociant: Illegal character in word
névé: Illegal character in word
od: Too short
ombré: Illegal character in word
or: Too short
orfèvrerie: Illegal character in word
outré: Illegal character in word
pari passu: Illegal character in word
pavé: Illegal character in word
piqué: Illegal character in word
plissé: Illegal character in word
poêlée: Illegal character in word
portière: Illegal character in word
précieuse: Illegal character in word
preterpluparenthetical: Too long
pro rata: Illegal character in word
pro tanto: Illegal character in word
pseudohermaphroditism: Too long
psychoneuroendocrinology: Too long
quevée: Illegal character in word
quinceañera: Illegal character in word
ratiné: Illegal character in word
recherché: Illegal character in word
récit: Illegal character in word
réclame: Illegal character in word
régisseur: Illegal character in word
relâche: Illegal character in word
repoussé: Illegal character in word
réseau: Illegal character in word
retroussé: Illegal character in word
roué: Illegal character in word
schwärmerei: Illegal character in word
se: Too short
soigné: Illegal character in word

soupçon: Illegal character in word
spectroheliokinematograph: Too long
sprachgefühl: Illegal character in word
sub dio: Illegal character in word
sub rosa: Illegal character in word
tabatière: Illegal character in word
tantième: Illegal character in word
totidem verbis: Illegal character in word
urdé: Illegal character in word
velouté: Illegal character in word
vergée: Illegal character in word
vivandière: Illegal character in word
xu: Too short
yu: Too short