Proceedings of the Seventeenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment
(AIIDE 2021)

A Demonstration of *Blabrecs*, an AI-Based Wordgame

Max Kreminski, Isaac Karth
University of California, Santa Cruz
{mkremins, ikarth}@ucsc.edu

Abstract

Blabrecs is an AI-based modification to the popular wordgame *Scrabble*. In *Blabrecs*, English dictionary words may not be played; instead, players may only play nonsense words that are approved by a classifier trained on a list of English dictionary words. Gameplay therefore revolves around inventing plausibly English-sounding nonsense words and learning how to fool the classifier. In this paper, we briefly introduce our design goals for *Blabrecs*; describe the process by which *Blabrecs* was designed; and present two distinct implementations of the game's AI judge.

Introduction

Blabrecs is an AI-based (Treanor et al. 2015) rules modification to the popular wordgame *Scrabble*. In *Blabrecs*, as in *Scrabble*, players take turns drawing letter tiles from a bag and placing these tiles on a grid to form words, which are then scored based on letter frequencies and tile score multipliers to award players with points. Unlike *Scrabble*, however, *Blabrecs* does not use an English dictionary to determine what letter sequences constitute valid words. Instead, it uses a classifier *trained on* the dictionary to accept or reject letter sequences. Actual dictionary words are disallowed; only nonsense sequences that the classifier misclassifies as words are allowed to be played.

Blabrecs is implemented as a web page[1]. It offers a text box into which words can be inserted for testing, plus an editable table of definitions for each valid word (so that players can build up a lexicon of imaginary words as they play) and a listing of the *Blabrecs* rules as they differ from those of *Scrabble*. The current version of *Blabrecs* offers two distinct classifiers: one based on a Markov chain and one based on a convolutional neural network. The players may switch between these classifiers freely; this allows them to develop a sense of how the classifiers differ in terms of what letter sequences they are likely to approve or disallow as words.

Since its initial release in December 2020, *Blabrecs* has been widely shared on social media. It has also been covered in several news outlets, including New Scientist[2]; The Next Web[3]; the board game news site Dicebreaker[4]; and the linguistics podcast Because Language[5]. This suggests to us that the game has resonated with audiences in at least some of the intended ways.

Design

We had several design goals for *Blabrecs*. First, we wanted to create a game that highlights the absurdity of trying to delineate English-language words as "valid" or not, whether by computational process or human-authored dictionary. We find *Scrabble*'s focus on dictionary memorization and flaky AI-based spelling checkers frustrating for similar reasons: in both cases, an external authority is imposed between the individual and their own language, often with alienating results. From this angle, *Blabrecs* can be viewed as a protest against the increasingly AI-mediated phenomenon of linguistic prescriptivism.

We also wanted to demonstrate how gameplay could be used to help players develop an intuitive feel for how an AI system works. In the course of *Blabrecs* gameplay, players are strongly incentivized to discover and exploit quirks in the AI gatekeeper's evaluation process; additionally, players can compare and contrast how words are evaluated by two different classifiers. As a result, players may come away from *Blabrecs* with a stronger intuitive sense of how their writing might be evaluated by the kinds of AI systems used here.

Finally, we wanted to create a game in which players build up a private language with one another as they play. In each *Blabrecs* play session, as players play new words, they are added to a table of player-editable definitions, allowing the players to collectively decide on meanings for the words they have invented. Some of these words may live on within the group of players as in-jokes, mirroring the way that a private lexicon is invented between the players in tabletop language creation games like *Dialect* (Thorny Games 2018).

To validate the high-level gameplay concept, the design of *Blabrecs* began with a Wizard of Oz prototype in which a

[1]https://mkremins.github.io/blabrecs

[2]https://www.newscientist.com/article/mg24933162-600-why-you-should-trust-reindeer-to-pick-stocks-over-politicians

[3]https://thenextweb.com/neural/2020/12/14/new-ai-scrabble-mod-only-allows-words-that-dont-exist

[4]https://www.dicebreaker.com/games/scrabble/news/scrabble-variant-nonsense-words-blabrecs-ai

[5]https://becauselanguage.com/17-words-of-the-week-of-the-year-2020

human played the role of the AI, judging letter sequences as valid or invalid on the basis of intuition. Several rounds of playtesting revealed that the invention of feasible nonsense words to bypass a gatekeeper agent could produce a compelling play experience, so we went forward with a computational version of the game.

Implementation

Markov Chain Classifier

The initial implementation of the *Blabrecs* classifier is based on a Markov chain trained on the ENABLE word list[6], which is often used as a baseline English dictionary for word games. To train the model, we first turn each word in the word list into a sequence of character trigrams; for instance, the word "apple" is turned into the sequence `["^ap", "app", "ppl", "ple", "le$"]` (where the ^ and $ characters represent the start and end of a word respectively). Then we calculate and store the frequency of each trigram relative to other trigrams that begin with the same two-character prefix.

To evaluate the plausibility of a letter sequence using this model, we divide it into a sequence of trigrams as before and look up the frequency of each trigram in the Markov chain. The per-trigram frequencies are first multiplied together to determine an overall likelihood score for the input letter sequence; this score will always be 0 if the sequence contains any trigrams that were not present within the ENABLE word list, and longer sequences will generally produce lower scores. Then we check whether this score is above or below the *average* likelihood score for real dictionary words of the same length. If the letter sequence is both more likely than the average real word of this length and does not appear in the dictionary, we allow it to be played.

This classifier is quirky. In particular, it can often be convinced to accept words that contain some highly implausible trigrams if several highly plausible trigrams are also present. Additionally, with the exception of the first and last trigram in each word, it pays no attention to where in the word a trigram occurs. Nonetheless, this classifier was the only one present when the game was launched, and it seems to mirror the typical player's intuitive sense of plausibility well enough to make for interesting play.

Neural Classifier

An alternative implementation of the *Blabrecs* classifier, provided by Isaac Karth, makes use of a convolutional neural network (CNN). This classifier is modeled loosely on the CNN-based text classifier presented in Step 4 of the Google Developers text classification guide (Google Developers 2021), but modified to work in TensorFlow.js (so that it can be used in a web browser) and to treat characters as tokens instead of words (since our goal, unusually for text classification, is to classify sequences of up to 16 letters, rather than longer passages of text).

Because ENABLE alone proved to contain too little data to train a good CNN, this classifier was instead trained on three word lists: the YAWL[7] (a strict superset of ENABLE), Letterpress[8], and Moby[9] word lists. These word lists were concatenated together, and duplicate words were removed. Additionally, we generated 2,016,000 unique non-word sequences of random letters between 3 and 24 letters in length to use as negative examples; this is approximately six times as many negative examples as there are positive examples in the combined word list. For this data generation task, we used a weighted random process to select letters at the same rate that they appear in known English words.

To evaluate a player-submitted letter sequence, we use the classifier to predict its likelihood of being a valid English word and check whether the predicted likelihood is greater than 0.82. This threshold was determined by manually testing a large number of words and picking a cutoff that seemed to match our intuitive notion of word plausibility.

Evaluating the quality of a nonsense word gatekeeper is difficult and largely intuition-driven. Altogether, though, the neural classifier seems to match the authors' intuition for nonsense word plausibility more reliably than the Markov chain classifier; in particular, it seems less prone to "false negatives", or judging nonsense words as implausible that the authors consider plausible. Additionally, the neural classifier's quirks are less obvious and easy to learn than those of the Markov chain classifier: it is more difficult to figure out what features the neural classifier weights most strongly in its estimation of nonsense word plausibility.

Related Work

In addition to the aforementioned language creation tabletop game *Dialect*, several other AI-based language games and explorations served as sources of design inspiration for *Blabrecs*. The *Scrabble*-like word construction game *Rewordable* (Parrish, Simon, and Szetela 2017) is of particular note for how the designers made use of AI to identify a set of letter sequences that could be used as cards to improve on *Scrabble*'s letter-tile-based gameplay, though in *Rewordable* the player does not interact directly with an AI system.

One of the first author's previous AI-based game projects—*Throwing Bottles at God* (Kreminski and Wardrip-Fruin 2018)—represents an earlier attempt to make Markov chains playable. Rather than classifying player-submitted text, *Throwing Bottles* makes use of Markov chains as a predictive text algorithm to help the player write short messages in a particular style; this can be viewed in hindsight as a failed experiment, whereas *Blabrecs* has been much more successful in eliciting the desired player experience.

Acknowledgements

Max Kreminski would like to thank their family for playtesting the initial Wizard of Oz prototype of *Blabrecs*.

[6] https://www.wordgamedictionary.com/enable

[7] https://github.com/elasticdog/yawl

[8] https://github.com/lorenbrichter/Words

[9] https://www.gutenberg.org/files/3201/files/SINGLE.TXT

References

Google Developers. 2021. Step 4: Build, Train, and Evaluate Your Model. https://developers.google.com/machine-learning/guides/text-classification/step-4. Accessed: 2021-08-05.

Kreminski, M.; and Wardrip-Fruin, N. 2018. Throwing Bottles at God: Predictive text as a game mechanic in an AI-based narrative game. In *International Conference on Interactive Digital Storytelling*, 275–279. Springer.

Parrish, A.; Simon, A.; and Szetela, T. 2017. Rewordable: The Uniquely Fragmented Word Game. https://rewordable.com. Accessed: 2021-08-05.

Thorny Games. 2018. Dialect: A Game About Language and How It Dies. https://thornygames.com/pages/dialect. Accessed: 2021-08-05.

Treanor, M.; Zook, A.; Eladhari, M. P.; Togelius, J.; Smith, G.; Cook, M.; Thompson, T.; Magerko, B.; Levine, J.; and Smith, A. 2015. AI-based game design patterns. In *Proceedings of the International Conference on the Foundations of Digital Games*.