Analyzing Transformer-based AI agents for Codenames

Zhong Ooi, Department of Computer Science
Mentor: Dr. Markus Eger
Kellogg Honors College Capstone Project for RSCA 2023

Introduction

Codenames, created by Vlaada Chvátil, is a word association game where a player is given 25 words and needs to generate a word and number pair to convey information that connects a subset of the given words to another group of players. This project is an extension of the "Cooperation and Codenames: Understanding Natural Language Processing via Codenames" by Kim et al., which uses Codenames as the medium to test the capabilities of natural language processing models. With a breakthrough in natural language processing through the creation of the transformer, the ability of natural language processing has increased dramatically.

Objective

- This project seeks to use this new technology to build an AI agent to play Codenames as either as a Codemaster or a Guesser
- By using codenames as a medium this project seeks to compared a transformer-based AI agent compared to Word2Vec and Glove.

Codenames

BOARD

CHAIR	HOLLYWOOD	DRESS	BUTTON	ENGINE
PIT	SLUG	TUBE	MASS	CARROT
BEAT	DROP	LEAD	BARK	WELL
TOWER	VET	PLATE	CAR	CIRCLE
LONDON	STATE	EGYPT	DANCE	MOUTH

Game Win Condition: Find all your team spies before the opposing team while dodging the assassin (purple color).

Game Rules
- If you choose a wrong word, your turn ends immediately
- Spymaster can not give clues if it contains a word on the board
- Teams are allowed to choose one extra word from the number given by the spymaster
- Guessers are allowed to end their turn early

Player Roles
- Codemaster: This player looks to his spies and generates a word that relates to several spies
- Guesses: This player uses the word and number generated by the spymaster to guess their spies

AI Agents

- **w2v**: "training approach that actually comprises two different approaches …the CBOW model takes the context as the input and tries to predict the word, while the Skip-Gram model takes the word and tries to predict the con-text[1]"
- **Glove**: "trained by linear regression that tries to learn weights such that the weights associated with a word try to predict the log of the occurrence counts of the word and its contexts [1]"
- **w2vglove**: "a concatenation of the vectors created by both w2v and glove [1] "
- **Transformer**: "sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention. [2]"

Transformer Codemaster Methodology

Codemaster

1. Generate a cosine similarity score between the board and all possible clues
2. Use mrjob, a map-reduce library, to condense and score all the data from step 1
3. Grab the word number pair from mrjob and rank them by score

Scoring Method

Mapper
- Map all the words and scores related to each clue

Reducer
- Rank all the words from given by the mapper
- Trim all words after the first instance of a non-red word
- Continue to trim words from the accepted list if they fall within a certain threshold when compared to non-red words
- Add all the points for each word associated with each clue after all the trimming
- Add the variance of the associated words for each clue

Transformer Guesser Methodology

Guesser

1. Generate a cosine similarity score between the board and clue
2. Return the highest similarity word
3. Restart from step 1 until the guesser guesses a wrong word or the number of guesses equals the number given by codemaster

Results

Average number of turns

Codemasters \ Guessers	W2V	Glove 300d	Glove 50d	W2VGlove 300d	W2VGlove 50d	Transformer	Transformer
W2V - Low	8.0	9.3	11.1	9.0	9.3	13.5	11.2
W2V - Med	7.9	9.2	10.9	9.0	9.2	13.4	11.0
W2V - High	5.4	7.8	8.6	6.8	7.1	14.5	8.4
Glove 300d - Low	11.4	8.0	8.6	8.1	8.7	16.5	12.3
Glove 300d - Med	8.8	7.8	8.3	8.1	8.0	16.4	12.2
Glove 300d - High	9.1	4.7	6.1	4.8	8.5	15.8	9.6
Glove 50d - Low	10.3	8.8	7.6	8.6	5.3	18.0	12.7
Glove 50d - Med	10.1	6.4	4.2	6.0	5.3	16.7	8.3
Glove 50d - High	7.9	5.8	3.3	5.6	4.0	17.7	9.4
W2VGlove 300d - Low	9.3	8.0	8.4	8.0	8.5	15.3	12.3
W2VGlove 300d- Med	9.2	8.0	8.3	7.9	8.5	15.1	12.1
W2VGlove 300d - High	6.5	5.0	5.8	4.9	8.4	15.6	9.3
W2VGlove 50d - Low	9.1	8.5	8.0	8.4	5.7	19.0	13.1
W2VGlove 50d- Med	7.4	6.7	5.8	6.4	4.9	19.2	10.6
W2VGlove 50d - High	6.2	5.3	4.0	4.9	3.4	18.2	9.4
Transformer Passive	12.8	12.8	16.0	11.7	14.9	5.0	
Transformer Aggressive	14.1	14.6	16.3	12.3	15.7	4.3	
Transformer Passive	8.4	8.3	10.1	7.7	9.9		
Transformer Aggressive	8.7	8.6	9.6	7.7	9.5		

*blue highlight is the average turn with all games
*red highlight is the average turn without assassin games

Minimum number of turns

Codemasters \ Guessers	W2V	Glove 300d	Glove 50d	W2VGlove 300d	W2VGlove 50d	Transformer
W2V - Low	8	8	8	8	8	8
W2V - Med	7	7	8	7	8	7
W2V - High	4	6	5	5	6	6
Glove 300d - Low	9	8	8	8	8	8
Glove 300d - Med	5	7	7	7	8	8
Glove 300d - High	5	3	4	4	7	6
Glove 50d - Low	8	6	6	6	4	7
Glove 50d - Med	7	4	3	4	4	6
Glove 50d - High	4	3	3	3	3	5
W2VGlove 300d - Low	8	8	8	8	8	8
W2VGlove 300d- Med	7	7	7	7	8	7
W2VGlove 300d - High	4	3	4	4	8	6
W2VGlove 50d - Low	7	7	7	7	4	8
W2VGlove 50d- Med	5	4	4	4	3	7
W2VGlove 50d - High	5	3	3	3	3	4
Transformer Passive	5	4	6	4	5	4
Transformer Aggressive	4	4	5	4	6	3

Win Percentage

Codemasters \ Guessers	W2V	Glove 300d	Glove 50d	W2VGlove 300d	W2VGlove 50d	Transformer
W2V - Low	100.0%	86.7%	73.3%	90.0%	80.0%	83.3%
W2V - Med	100.0%	86.7%	73.3%	90.0%	80.0%	83.3%
W2V - High	100.0%	76.7%	56.7%	86.7%	73.3%	63.3%
Glove 300d - Low	76.7%	100.0%	90.0%	100.0%	96.7%	66.7%
Glove 300d - Med	66.7%	100.0%	90.0%	100.0%	93.3%	66.7%
Glove 300d - High	56.7%	100.0%	90.0%	100.0%	93.3%	60.0%
Glove 50d - Low	86.7%	100.0%	100.0%	93.3%	73.3%	56.7%
Glove 50d - Med	86.7%	83.3%	100.0%	73.3%	80.0%	50.0%
Glove 50d - High	76.7%	83.3%	100.0%	73.3%	90.0%	46.7%
W2VGlove 300d - Low	93.3%	100.0%	90.0%	100.0%	100.0%	76.7%
W2VGlove 300d- Med	93.3%	100.0%	90.0%	100.0%	100.0%	76.7%
W2VGlove 300d - High	93.3%	100.0%	83.3%	100.0%	100.0%	60.0%
W2VGlove 50d - Low	90.0%	96.7%	96.7%	93.3%	86.7%	50.0%
W2VGlove 50d- Med	86.7%	96.7%	96.7%	93.3%	83.3%	40.0%
W2VGlove 50d - High	66.7%	73.3%	93.3%	80.0%	100.0%	43.3%
Transformer Passive	73.3%	73.3%	60.0%	76.7%	66.7%	100.0%
Transformer Aggressive	66.7%	63.3%	56.7%	73.3%	60.0%	100.0%

Analysis of a game from the AI transformer Codemaster

This game was chosen due to the clear areas of improvements and a flaw with the current implementation

BOARD

BUTTON	BLOCK	CANADA	THIEF	BEAT
BANK	MUG	LINE	NINJA	COURT
IVORY	LONDON	KID	TIME	COLD
ORANGE	MILLIONAIRE	TIE	OLIVE	GREEN
YARD	BEACH	PART	FISH	LIFE

Codemaster: Transformer
Guesser: W2VGlove (300D)
SEED: 1950

#	Codemaster Clue			Guesser Answer
1	Money	3	LIFE,BANK,TIME	TIME,BANK,PART
2	CHILDHOOD	2	KID,LIFE	LIFE,KID
3	NORWEGIAN	2	CANADA,NINJA	CANADA,LONDON
4	JENNIFER	2	NINJA,MILLIONAIRE	BEAT
5	JENNIFER	2	NINJA,MILLIONAIRE	GREEN
6	JENNIFER	2	NINJA,MILLIONAIRE	BEACH
7	JENNIFER	2	NINJA,MILLIONAIRE	COURT
8	JENNIFER	2	NINJA,MILLIONAIRE	BUTTON
9	JENNIFER	2	NINJA,MILLIONAIRE	MILLIONAIRE,TIE
10	SUZUKI	1	NINJA	THIEF
11	SUZUKI	1	NINJA	NINJA
12	DRAWN	1	LINE	LINE

Summary and Conclusions

Transformer codemaster facing transformer Guesser
- It performs exceptionally well against a transformer guesser with a low avg guess.
- No fail games between these two AI agents

Transformer codemaster facing non-transformer Guesser
- The transformer codemaster finds the assassin far too often, leading to a higher avg guess percent and a lower win percent.
- The average number of turns without assassin loss shows that the AI performance is significantly brought down due to its low win rate
- The min guesses and Average amount of turns without assassin show that the transformer has the potential to perform as well as the other model's implementations

Non-transformer codemaster facing transformer Guesser
- The transformer guesser finds the assassin far too often, leading to a higher avg guess percentage and a lower win percent.
- The average number of turns without assassin loss shows that the AI performance is significantly brought down due to its low win rate
- The minimum guesses is consistent with the other AI, meaning if optimized, the transformer could perform significantly better

Fixes to improve the interaction between the transformer codemaster and non-Transformer Guesser
- Employ a method to ensure the same clue isn't given multiple times or in succession.
- Place a harsher penalty on incorrect words when generating the score.

Future Work

- Improve and fine-tune the transformer codemaster scoring method, as explained in the summary and conclusions.
- Improve the implementation of the game by adding the opposing team into the game. This would allow us to test the AI in an adversarial environment where guessing the opposing team's spies is highly detrimental.
- Create a UI for the game, so it is humans can play against the AI in a more elegant way than the current command-line implementation.
- Test each AI against humans to see the interaction between each NLP model and its effectiveness in a real-world application.

Acknowledgments

I want to thank Dr. Adam Summerville for recommending me this project when I was unsure what to work on and for helping me find a new mentor to finish this project.

I would also like to thank Dr. Markus Eger for picking up this project in the middle of the summer and supporting me throughout this whole process while providing me invaluable feedback.

References

[1] A. Kim, M. Ruzmaykin, A. Truong, and A. Summerville, "Cooperation and codenames: Understanding Natural Language Processing via codenames," Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 15, no. 1, pp. 160–166, 2019.
[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. arXiv. https://doi.org/10.48550/arXiv.1706.03762
[3] all-MiniLM-L6-v2, Sentence Similarity, https://www.sbert.net