Language emergence with embeddings from a language model

Dewi E. Timman
12419273

Bachelor thesis
Credits: 12 EC

Bachelor *Cognition, Language and Communication*

University of Amsterdam
Faculty of Humanities
Spuistraat 134
1012 VB Amsterdam

Supervisor
Dr. R. G. Alhama

Institute for Logic, Language and Computation
Faculty of Science
University of Amsterdam
Science Park 107
1098 XG Amsterdam

October 16, 2024

Abstract

Compositionality is an important characteristic of natural language. It allows us to make infinitely many expressions from a finite amount of building blocks. In this research, two agents take part in a referential communication game. In such a game, agents try to communicate in order to achieve a goal. Where previous research focused on agents without pre-linguistic knowledge, this one focuses on agents with pre-linguistic knowledge. The pre-linguistic knowledge consists of word embedding vectors from large language model BERT. In order to measure what the effect is of structure that is already present in these word embedding vectors, two versions of the referential communication game are made. The first version takes images as input and the second version word embedding vectors. Results show that in both models a language does emerge for communication. However, only the language emerging from the image model shows signs of the use of compositionality.

Keywords: language emergence, compositionality, large language models, artificial intelligence, computational linguistics, referential communication game, reinforcement learning

The code of this work can be found at GitHub, https://github.com/dewi-elisa/Thesis-CLC

Contents

Chapter 1

Introduction

We are able to talk and communicate with each other. However, natural languages are very complex. A characteristic of natural language that makes it complex is ambiguity. While ambiguity refers to words with multiple meanings, Piantadosi, Tily, and Gibson (2012) suggest ambiguity causes greater communicative efficiency. This is the case because there are fewer words needed in a language because usually the right meaning can be chosen from the context. Another characteristic that allows us to communicate more efficiently is compositionality. According to Frege (1892), compositionality is the notion that the meaning of the whole is made up from the meanings of its constituents. It thus allows us to understand concepts we have not seen before. For example, if we know what a 'green car' and a 'purple coat' are, we know what 'purple car' would be because it consists out of two words we saw before. So, we know a lot about how natural languages look like and how they work. But, what is still unclear is, how did language originate in the first place? And how did it become so complex? This is exactly what language emergence investigates.

1.1 Language emergence

The area of language emergence builds on top of that of game theory and philosophy. It is currently of interest in machine learning (ML), natural language processing (NLP), linguistics and cognitive science. In order to do so, often linguistic communication games are used. But why?

Wittgenstein (1953) argued that language derives meaning from its use. Therefore, by playing a language game, language is used and meaning derived. Thus, a language game would be an excellent candidate to study how language originated. In addition, Austin and Urmson (1962) argue that words are used by humans to coordinate with others and make things happen. Thus, communication also plays

a role. Therefore, a linguistic communication game would help research language emergence.

In a linguistic communication game, agents try to communicate with each other. Usually, agents do this without any prior linguistic knowledge. The agents try to solve a task together and receive a reward if they succeed. To see if a human-like language emerged, usually the degree of compositionality is measured. Another measure is generalizability. A few other measures exist, but more research is needed to determine what good evaluation metrics are (Boldt & Mortensen, 2024). In this research, however, we look at compositionality.

The current research distinguishes itself from previous research by giving agents some linguistic knowledge. The goal is to investigate whether the agents do something with this knowledge.

Chapter 2

Literature background

To understand the current research, some more understanding about agent-based linguistic communication, referential communication games, compositionality, neural networks and word embedding vectors is needed. This will be provided in the following sections.

2.1 Agent-based linguistic communication

Traditional natural language learning approaches take large text corpora and try to infer structural properties of language (Radford et al., 2019). Therefore, communication is not taken into account. In this research, there will be looked at agent-based linguistic communication, which does take communication into account. An agent is "An entity having the property of controlling its' own action independent of other entities unless it desires to communicate with other entities. It typically does not refer to a second party unless it possesses insufficient knowledge to perform that particular action itself." (Ingham, 1997, p.14). An agent can therefore be a human, a robot or a computational model. In the case of this research, it is the last one. Moreover, the agent wants to learn a language in order to communicate. This means that there is communication between two or more agents. Usually, that means the agents need to solve a task together. To solve the task, agents need to successfully communicate to each other.

Agent-based linguistic communication does not always follow natural language. Chaabouni, Kharitonov, Dupoux, and Baroni (2019) found that emergent communication does not always adheres to Zipf's law of Abbreviarion, whereas Kottur, Moura, Lee, and Batra (2017) found that emergent communication does not always follow the compositionality patterns of natural language. Moreover, Lazaridou and Baroni (2020) discusses different types of communication. Communication can be continuous (for example, the DIAL system in Foerster, Assael, de Freitas,

and Whiteson (2016)) or discrete (for example, the RIAL system in the same research). Lazaridou and Baroni (2020) and Boldt and Mortensen (2024) both provide overviews of emergent communication.

Lewis (1969) first proposed a signaling game to research communication. This game is adapted many times for different purposes. An overview can be found in Zubek, Korbak, and Rączaszek-Leonardi (2023). One of those adaptations is the referential communication game.

2.2 Referential communication games

To research this agent-based linguistic communication, agents can participate in a referential communication game. In such a game, agents talk about objects or other entities in a specified world. To do this, they need to come up with a language to communicate about their world. Usually, the agents do not have any prior linguistic knowledge. A more formal definition of the referential communication game can be found in section 3.2.

The game can also be adjusted. For example, the game can be constrained by messages of just one symbol (Lazaridou, Peysakhovich, & Baroni, 2017) or by using multiple symbols (Lazaridou, Hermann, Tuyls, & Clark, 2018). In addition, the game can be played for multiple turns (Evtimova, Drozdov, Kiela, & Cho, 2017). The goal of the task can be adjusted (Bouchacourt & Baroni, 2019), or multiple communities of agents can be used (Graesser, Cho, & Kiela, 2019). Moreover, the emerged language can be grounded (Lazaridou et al., 2017) so that it is understandable for humans. Lastly, the referential communication game can be used for other tasks such as machine translation (Y. Li, Ponti, Vulić, & Korhonen, 2020) or played with robots (Deichler, Wang, Alexanderson, & Beskow, 2023).

2.3 Compositionality

The emerged language from the referential communication game can be investigated. One characteristic of natural language that can be investigated in emerged lanuages is compositionality. According to Frege (1892), compositionality is the notion that the meaning of the larger is determined by the meaning of its parts and how they are put together. For example, a language with no compositionality, would have two totally different words for the notions 'green car' and 'red car'. The overlap created in English with the word 'car' means that there is compositionality present. Because of compositionality, infinite expressions can be allowed through a finite dictionary and a finite set of combination rules (Lazaridou et al., 2018). Previous work has shown that agents can produce (somewhat) compositional lan-

guage when playing language games (Steels, 2003). Furthermore, Lazaridou et al. (2018) find that the degree of structure found in the input data influences the degree of compositionality in the emerged language. The more structure in the input data, the more likely it is for a structured compositional language to emerge. Lastly, Chaabouni, Kharitonov, Bouchacourt, Dupoux, and Baroni (2020) argue that the more compositional a language is, the more easily it is picked up by other agents, even if they have a different architecture.

Now that we know what compositionality is, there is the question of how we can measure it. To measure the degree of compositionality of a language, different metrics are proposed in Chaabouni et al. (2020). Some researchers (Kottur et al., 2017; Cogswell, Lu, Lee, Parikh, & Batra, 2019) look at how well the emerged language can be generalized: how well does the language work when there are inputs that have not been seen before? Other researchers (Brighton & Kirby, 2006; Lazaridou et al., 2018; F. Li & Bowling, 2019) use topographic similarity. This is also the metric used in this research since this has become the standard in language emergence research (Chaabouni et al., 2020). More information about topographic similarity can be found in section 4.3. Two more metrics for compositionality based on juxtapositions are proposed in Chaabouni et al. (2020). They will not be discussed here since they are of no interest for this research.

2.4 Neural networks

In order to build the agents in the referential communication game, three types of neural networks will be used. Each of the networks have their own advantages. They will be further discussed in the following sections.

2.4.1 Multilayer perceptron

A multilayer perceptron (MLP; Rosenblatt, 1958) is one of the most simple types of neural networks. An overview can be seen in figure 2.1. It is fully connected, which means that, from two connected layers, from every node (represented by circles) of the first layer, to every node of the second layer, a connection is made (represented by arrows). The nodes can be seen as neurons and all connections have weights, reflecting the strength of that connection. There are three types of layers: an input layer, a hidden later, and an output layer. The input and output layers consist of nodes with the input and output, respectively. The hidden layer processes the inputs and converts them to the outputs. An MLP can have multiple or no hidden layers.

Thus, an MLP converts an input into an output. The amount of inputs does not need to match the amount of outputs. Therefore, an MLP can be used for

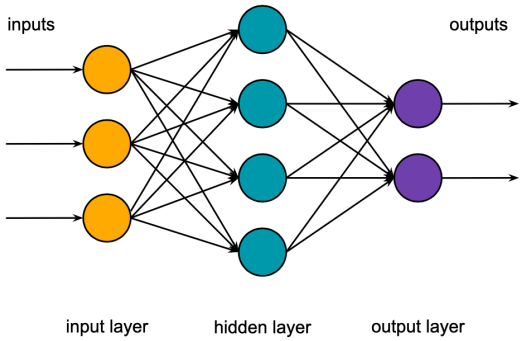


Figure 2.1: Schematic overview of an MLP. The orange nodes represent the input layer, the blue nodes the hidden layer and the purple nodes the output layer.

converting data to a smaller size, i.e. making a dense representation. To this network type is referred as linear layer in chapter 3. In this case, the MLP does not have any hidden layers. Moreover, an activation function is used. An activation function transforms the inputs of a neuron into an output determining if the neuron should be activated or not. In this research, the hyperbolic tangent function (tanh; Szandała, 2020) is used.

2.4.2 Long-short term memory

A Long-Short Term Memory (LSTM; Hochreiter & Schmidhuber, 1997) is a type of recurrent neural network (RNN). This means that the input goes through the network one by one, instead of all at once (like with an MLP), i.e. word by word. This makes it possible for the model to process information with different kinds of lengths, i.e. sentences with different amounts of words where all words are transformed into vectors with the same length. Moreover, a LSTM takes previous input into account. Therefore, it takes some context into account making it a great option for generating language.

2.4.3 Convolutional neural networks

A Convolutional neural network (ConvNet; LeCun, Bengio, & Hinton, 2015) has one or more convolutional layers. A convolutional layer has a filter (or kernel) that slides over the input data. This kernel is learned during the training of the network. Therefore, this network is especially good at extracting certain features in images. The network can also have multiple kernels to extract more information. The amount of kernels is called channels. The kernel can slide over the input one-by-one, or it can skip some inputs. This is called the stride. A stride of 1

means that no inputs are skipped. After a convolutional layer, often an activation function is used. It is common to use the rectified linear unit activation function (ReLU; Fukushima, 1969).

2.5 Word embedding vectors

The input of a neural network is often a vector. This vector encodes certain information in a numerical manner. In this research, the vector contains information about words, it is a word embedding vector. A word embedding vector is a word that is represented as a vector using numbers. For example, the word 'car' can be represented by the vector $\begin{bmatrix}1 & 7 & 4\end{bmatrix}$. This is also how Large Language Models (LLMs) represent text. LLMs learn the word embedding of a word by looking what other words appear in its context. Therefore, the more similar two words are, the closer they are in the vector space. Thus, there is a certain structure present in these word embedding vectors.

2.5.1 BERT

Bidirectional encoder representations from transformers (BERT) is a LLM made by Google (Devlin, Chang, Lee, & Toutanova, 2018). BERT is based on the transformer network architecture (Vaswani et al., 2017). Transformers are based on attention (Bahdanau, Cho, & Bengio, 2014). Attention determines the importance of a word in a sentence relative to the other words in that sentence. Transformers therefore do not have any recurrent units, which is the case by, for example, LSTMs. Because of this, transformers have a shorter training time than LSTMs.

BERT is used as a baseline in many natural language processing tasks (Rogers, Kovaleva, & Rumshisky, 2020). The original versions have 110 and 340 million parameters, which is quite large. Therefore, a lot of smaller, different versions are available nowadays, for example, ALBERT (Lan et al., 2019). In addition, BERT is also available for other languages than English, for example, the Dutch BERTje (de Vries et al., 2019).

Rogers et al. (2020) researched what BERT actually seems to know. They found that BERT embeddings contain information about syntax, e.g. part of speech and syntactic roles. Moreover, BERT encodes semantic knowledge such as semantic roles and entity types (Tenney, Xia, et al., 2019). In this research, BERT is used to extract embeddings from the labels of labeled images.

2.6 Current research

In section 2.2 we saw that previous research on language emergence with referential communication games focussed on agents without any pre-linguistic knowledge. In this research, however, the agents are given pre-linguistic knowledge in the form of a word embedding vector from BERT. The question is whether the agents use the structure that is already present in the word embedding vectors. In other words, what is the effect of word embedding vectors from a language model on the emergence of agent-based linguistic communication from referential communication games?

In order to answer this question, two versions of the referential communication game are designed. One version takes images as input, the other the word embedding vectors. The game will be evaluated to see if the agents communicate successfully and if they do so using compositionality. If the agents do communicate successfully this means a language is emerged between the agents. Moreover, if this language uses compositionality, the emerged language is human-like.

Chapter 3

Method

To research what the effect is of word embeddings on language emergence, two models are made: one model that has images as input, and one that has word embeddings as input. The method is based on Lazaridou et al. (2018).

3.1 Data

A downsampled ImageNet dataset with an image size of 64×64 pixels with 3 color channels is used (Chrabaszcz, Loshchilov, & Hutter, 2017). The training dataset, which is used for this research, consists of 1,281,167 images with 997 different labels. From this dataset, 20,000 games are created by randomly sampling 5 items (1 target item and 4 distractor items) from the dataset. To ensure that there are no duplicates in the word embedding model, games that have 2 or more equal labels are resampled. The dataset is then split into a train, validation and test set with 12,000, 4,000 and 4,000 games, respectively. For each model, the dataset is further adjusted to the needs of that model. These adjustments and more detail about the datasets are discussed in sections 3.3.1 and 3.4.1.

3.2 Experimental design

In this research, agents participate in a referential communication game. The game is adapted from Lazaridou et al. (2018) and is a variant of the Lewis signaling game (Lewis, 1969). More formally, the game works as follows. There is a set of items. From this set, K items are drawn at random. One of those items is the target t. The other items are distractor items, $D = \{d_1, \ldots, d_{K-1}\}$. Together, these are the candidate items, $C = \{t \cup D\}$.

In addition, there are two agents: a sender and a receiver. The sender can see

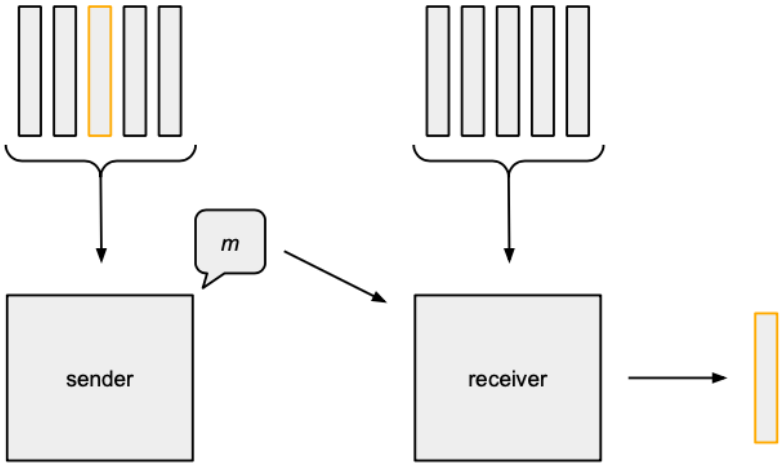


Figure 3.1: Schematic overview of the referential communication game. The item with the orange border represents the target item.

t, while the receiver sees C. However, the receiver does not know which of the items is the target t.

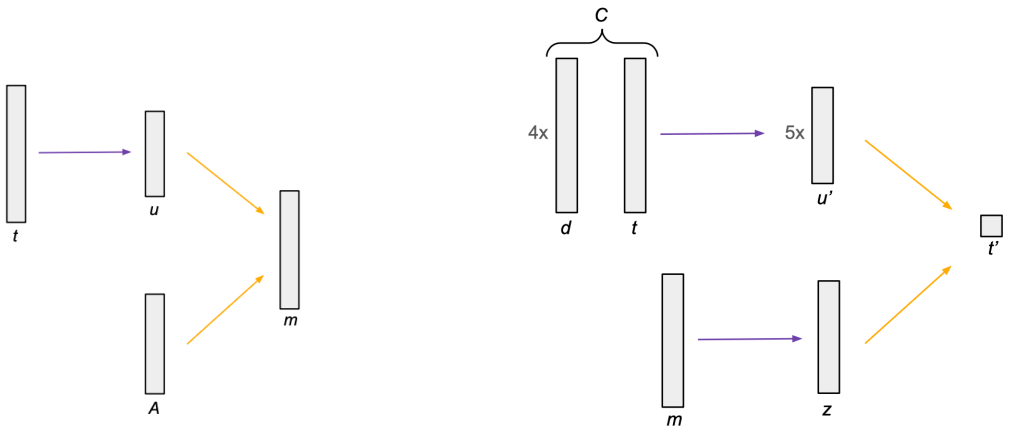
Moreover, there is an alphabet A of size N. The sender chooses one or more symbols to describe t. This is called the message m. The receiver does not know the target. However, it does see the message from the sender. The receiver then tries to guess the target image based on the message. This is the predicted target, t'.

If the receiver guesses the right target, so if $t = t'$, the agents achieve communicative success. In return, the agents get a reward. If no communicative success is achieved, the agents do not receive a reward. Based on this reward, the agents learn how to communicate with each other. A schematic overview of this game can be seen in figure 3.1.

3.2.1 Model architecture

The model consists of a sender and a receiver. The only connection between the two agents is the message m that is generated by the sender. Both agents consist of an encoder-decoder model. An overview can be seen in figure 3.2. The model is programmed in Python 3.12.2, using the EGG library (Kharitonov, Dessì, Chaabouni, Bouchacourt, & Baroni, 2021).

Sender. The sender receives the target item t and alphabet A and outputs the message m. To do this, it encodes the target item into a dense representation u. Then, u, together with the alphabet A, is decoded into m. See figure 3.2a for a



(a) sender (b) receiver

Figure 3.2: Schematic overview of the data representations in the agents. Purple arrows indicate processes that are part of an encoder, while orange arrows indicate processes that are part of a decoder.

schematic overview and sections 3.3.2 and 3.4.2 for more detail about the encoder and decoder for each model.

Message. The message consists of one or more words from A. The message is generated by sampling symbols from A until a stop symbol is generated or the maximum length L is reached. The words in A do not have any a priori meaning. Therefore, A can be seen as a list of possible words that the agent can use, but need to define themselves.

Receiver. The receiver receives the message m, and the set of candidate items C and outputs a predicted target t'. In order to do so, the agent first encodes each candidate item into a dense representation $u' \in U'$. In addition, m is encoded into another representation z. Finally, U' and z are decoded, and the receiver determines the predicted target item t'. See figure 3.2b for a schematic overview and sections 3.3.2 and 3.4.2 for more detail about the encoders and decoder for each model.

Communicative success. When the predicted target t' is equal to the actual target t, communicative success is achieved. The agents are then rewarded. If t' is not equal to t, the agents do not get a reward. More formally,

$$R = \begin{cases} 1, & t = t' \\ 0, & \text{otherwise,} \end{cases}$$

where R is the reward function. The parameters of the model are estimated using the `REINFORCE` update rule (Williams, 1992) with an Adam optimizer.

Hyperparameters. In both models, the agents use a dense representation with size 50. The learning rates of both the sender and the receiver are 0.001 and the Adam optimizer has a learning rate of 0.0001. Moreover, a batch size of 32 is used. For the image model 25 epochs are used and for the word embedding model 50. In the referential game, 4 distractor items are used. In addition, two versions of the game are run for both models, one with an alphabet size of 100 and a maximum message length of 10 and one with an alphabet size of 17 and a maximum message length of 5. This in in line with earlier research from Kottur et al. (2017) suggesting that non-compositional language emerged in the case of overcomplete alphabets.

3.3 Model 1: images

In the first model, the data used consists of images.

3.3.1 Image data

The dataset described in section 3.1 is further adjusted. For this model, only the images are needed. Therefore, the labels are removed from the dataset. All images are represented by a tensor (a multidimensional matrix) with shape $64 \times 64 \times 3$.

3.3.2 Image model

As described in section 3.2.1, the model consists of two agents which are both encoder-decoder models. In this section, a more detailed description of those encoders and decoders will be given.

Sender. The sender consists of an encoder that converts the target t into a dense representation u and a decoder converting u and alphabet A into a message m (see figure 3.2a). The encoder consists of an 6-layer convolutional neural network (ConvNet; LeCun et al., 2015) followed by a linear layer with the hyperbolic tangent activation function (Szandała, 2020). The ConvNet is not pre-trained so that the agent gets its own representations from the image. The ConvNet has 32 filters, a kernel size of 3, batch normalization, and strides [1, 2, 1, 2, 1, 2] for each layer. The decoder consists of a single-layer long short-term memory (LSTM; Hochreiter & Schmidhuber, 1997).

Receiver. The receiver consists of an encoder that encodes the items in C into dense representation U', an encoder that encodes the message m into representation z and a decoder that takes both U' and z to determine a target t'. The first encoder, to make dense representations U' from C, has the same architecture as the encoder of the sender. However, no weights are shared between the sender and the receiver. Therefore, the encoders do have the same architecture, but are not exactly the same.

The second encoder makes a representation z from the message m using a single-layer LSTM (Hochreiter & Schmidhuber, 1997).

Finally, the decoder takes the dot product between all encoded items in U and z. It then samples the target object t' from a Gibbs distribution calculated with this dot product.

3.4 Model 2: word embeddings

In the second model, the data used consists of word embeddings.

3.4.1 Word embedding data

The dataset described in section 3.1 is further adjusted. For this model, only the labels are needed. Therefore, the image part of the dataset is removed. Furthermore, the labels are given to the BERT-base model (Devlin et al., 2018) to obtain word embeddings of the labels. This is done as a preprocessing step rather than a step inside the model, since we are not training the BERT model. Therefore, this is a deterministic process for all labels and this significantly speeds up the training of our model. The word embeddings of the BERT model are represented as a 768-dimensional vector.

3.4.2 Word embedding model

The word embedding model is almost the same as the image model described in 3.3.2. The only difference is the architecture of the encoders. In the word embedding model, the encoders consist of only the linear layer followed by a tanh activation function instead of a ConvNet followed by the linear layer and the activation function.

Chapter 4

Results

To evaluate the models, multiple metrics are used. First, lexicon size is used to see if the messages send differed from each other. Second, accuracy is used to see how well the communication between the agents is: did the agents create a language? Last, compositionality is used to measure how human-like the language is. The results on the validation set can be seen in table 4.1 and on the test set in table 4.2. The test set has 3992 different images and 562 different word embeddings. Both models were evaluated two times with different values for the alphabet size and maximum message length. We will also take a look at the loss of the models. The results will be further explained in the following sections.

4.1 Lexicon size

The lexicon size is the number of unique messages used when testing the model. It can be calculated by calculating the length of the subset of messages. A high lexicon size means that there are a lot of different messages used. A low lexicon size would mean a lot of messages are equal to each other.

model	max length	alphabet size	accuracy	topsim
image	10	100	76.9 %	0.201
	5	17	76.4 %	0.237
word embedding	10	100	86.0 %	0.055
	5	17	83.1 %	0.054

Table 4.1: Results of both models on the validation set. The topographic similarity metric is represented by 'topsim'.

model	max length	alphabet size	accuracy	topsim	lexicon size
image	10	100	77.1 %	0.195	73
	5	17	75.1 %	0.227	45
word embedding	10	100	86.8 %	0.055	56
	5	17	83.6 %	0.052	40

Table 4.2: Results of both models on the test set. The topographic similarity metric is represented by 'topsim'.

In table 4.2 it can be seen that the image model used more distinct messages than the word embedding model. Namely, 73 and 45 for the image model and 56 and 40 for the word embedding model respectively. Moreover, the lexicon size is bigger when the alphabet is bigger and the maximum message length is shorter.

4.2 Accuracy

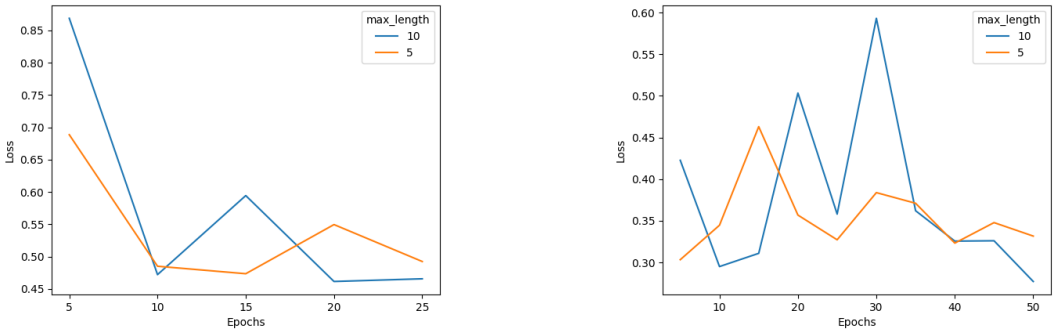
A low accuracy would mean that the agents fail to communicate well. From this, we could conclude that no language-like system emerges between the two agents. A high accuracy, however, would indicate that the agents are able to communicate and therefore created a language together. Accuracy is represented as a percentage and is calculated by:

$$\text{accuracy} = \frac{\#\text{ items where } t = t'}{\text{total }\#\text{ items}} \cdot 100.$$

When looking at table 4.1 and table 4.2, we see a higher accuracy for the word embedding model than the image model. In addition, the accuracies with a higher alphabet size and maximum message length are higher. Lastly, the accuracies are around 80%.

4.3 Topographic similarity

Topographic similarity is used to measure compositionality. It is represented by a number between -1 and 1. Here, a value of 0 means there is no compositionality at all. The more topographic similarity goes into the direction of -1 or 1, the more compositional the language is. Applying the definition of Brighton and Kirby (2006) to our situation, topographic similarity can be defined as the correlation of the distances between all the possible pairs of meaning (target items) and their corresponding message. The topographic similarity is therefore calculated by



(a) image model

(b) word embedding model

Figure 4.1: The loss of both models every 5 epochs on the validation set.

taking the distances between all pairs of sender inputs and the distances between all pairs of messages. Then, the negative Spearman correlation is taken.

In table 4.1 and table 4.2 all topographic similarity scores are positive. Moreover, a low topographic similarity score can be seen for the word embedding model. However, there is a higher score for the image model. In addition, we can see that when increasing the maximum message length and the alphabet size, the topographic similarity for the decreases for the image model but increases for the word embedding model.

4.4 Loss

In figure 4.1 the loss of both models can be seen. Loss is an important metric when training neural networks. It measures the error that the model has. In other words, the fewer mistakes the agents make, the lower the loss. When training a model, usually in the first epochs, the loss goes down. After some time, the loss stabilizes and does not go down a lot anymore. When the loss stabilizes, the model does not improve a lot anymore and the model can stop training. We can see that for both models, the loss is going down. Note that the overall difference in loss (so the y-axis) is not very big in general.

Chapter 5

Conclusion

To see what the effect of word embedding vectors is on language emergence in a referential communication game multiple versions of the referential communication game were made. These versions are compared in the following ways: lexicon size, accuracy and topographic similarity. First, there is the question whether the agents do communicate succesfully. Second, there is the question whether the emerged language is compositional. These questions are answered in the following sections.

5.1 Lexicon size

If we compare the lexicon size with the amount of games in the test set, we can see that a lot of games got the same message. Moreover, if we take into account that the sender had 3992 different images and 562 different word embeddings as input, we can see that multiple target items generated the same message. Thus, these results indicate a high sense of ambiguity. This only becomes more evident when looking at the difference in lexicon size when comparing the same model with the different maximum message lengths and the alphabet sizes.

5.2 Accuracy

All accuracies are higher than that of a random baseline (20%). This indicates that the agents did achieve successfull communication. In addition, a higher maximum length gives a higher accuracy, which is in line with earlier research of Skyrms (2010, p.131) and Lazaridou et al. (2018). This also makes sense from a logical point of view: the more symbols an agent can send the more distinct messages it can send. This is because the amount of possible messages is N^L, where N is the size of the alphabet and L the maximum message length. Therefore, the number

of distinct messages is dependent on the maximum length.

5.3 Topographic similarity

The results indicate that in the image model, some similar objects receive similar messages. However, in the case of the word embedding model, this is not the case. An interesting finding is that the topographic similarity increases for the word embedding model while it decreases for the image model when increasing L and N. However, this increase for the word embedding model is very small. If we look at the amount of distinct inputs of both models, we also see a big difference. This difference might explain the differences in topographic similarity for both models.

5.4 Loss

Figure 4.1a indicates the loss stabilized, and no more training was needed for the image model. Figure 4.1b indicates the word embedding model could have benefited from some more training, since the loss is still going down and did not yet stabilize.

5.5 The emerged language

To conclude, a language did emerge when using word embeddings as well as images. Nevertheless, the degree of compositionality in both models is different. The agents do not do something with the structure already present in the word embedding vectors of BERT. However, for the image model a more compositional language does emerge.

Chapter 6

Discussion

So for both models, a language emerged. However, only the image model showed clear signs of the use of compositionality in the language. The use of pre-linguistic knowledge in the agents did cause a more effective language. However, the language showed few signs of compositionality as is demonstrated by the low topographic similarity score. Some limitations followed by directions for feature research are discussed in the following sections. They are broken down into three areas: the language model used, the dataset and the model. But first the image model is compared with the one in Lazaridou et al. (2018).

6.1 Image model comparison

Since this research is based on Lazaridou et al. (2018) and both their research and this research have an image model, these can be compared. An image of the results of Lazaridou et al. (2018) can be found in figure 6.1 for easy reference. Note that the circumstances in Lazaridou et al. (2018) are different: there are either 2 or 20 distractors and a different dataset was used. Despite these differences, it can be seen that the accuracies in figure 6.1 are higher than the ones of this research. However, the topographic similarity in this research is higher. These findings can be due to the difference in datasets. The research of Lazaridou et al. (2018) namely used a dataset with images depicting an object. Each object has a specific shape, color, background and location. Therefore, there were fewer features present in the image making it easier to classify but more difficult to use compositionality.

game	distractors	balanced	viewpoints	lexicon size	random	train	test	topographic ρ
A	20	No	No	1068	5.0	93.7	93.6	0.13
B	2	No	No	13	50.0	93.2	93.4	0.006
C	2	No	Yes	8	50.0	86.0	85.7	0.07
D	2	Yes	Yes	5	50.0	90.4	89.9	0.06

Figure 6.1: Results from Lazaridou et al. (2018). The 'balanced' and 'viewpoints' columns are of no interest to us. The 'random', 'train' and 'test' columns represent accuracies on the respective datasets. Topographic similarity is denoted as 'topographic ρ'.

6.2 LLM

As discussed in section 2.5.1, BERT embeddings have some structure. In this research the embedding came from the last layer in the BERT model. However, research shows that certain layers of the model have certain functions. Ethayarajh (2019) shows that the last layers of the BERT model are more context-specific. Moreover, Rogers et al. (2020) find that multiple studies with different tasks, datasets and methodologies all are in consensus that syntactic information is most prominent in the middle layers of BERT. In addition, Tenney, Das, and Pavlick (2019) suggest that, contrary to syntactic information, semantic information appears to be encoded throughout the entire model. Lastly, Rogers et al. (2020) also reports multiple studies found that lexical semantic information is better encoded in distilled contextualized embeddings.

Because of these findings, more research can be done to see if an embedding vector from a different layer from the BERT model has more effect on the use of compositionality in the emerged language. Moreover, a different version of the BERT model can be chosen, or a totally different LLM that encodes semantic information better, since the agents in this research are mainly interested in semantics.

6.3 Dataset

As mentioned in chapter 4, the number of distinct inputs for the image model (3992) was a lot higher than for the embedding model (562). This can have influenced the comparability of the results. Future research can therefore use a different dataset in which this difference is smaller.

In addition, the BERT model takes a concept, which can consist of multiple words. There might be more structure present in a word embedding from one word only. Therefore, future research can give only one word to the model. For example,

the labels 'Egyptian cat' and 'Persian cat' can be changed to 'cat'. However, this way the degree of compositionality in the dataset decreases. Therefore, the degree of compositionality in the emerged language can also decrease as discussed in section 2.3.

6.4 Model

According to Vogt (2005), a transmission bottleneck can give rise to the emergence of compositionality. In other words, limiting the maximum message length or the alphabet size can cause to emerging language to use more compositionality. When looking at the topographic similarity of both models (see table 4.1 and table 4.2), we can see the compositionality increasing for the word embedding model when we limit the alphabet size and the maximum message length. Because of this, the word embedding model might benefit from further limiting those parameters.

Moreover, since designing neural networks is more an art than a science (Erenshteyn, Foulds, & Galuska, 1994), different model configurations can be tried. For example, the size of the dense representation can be adjusted, or the size of the hidden cells from the LSTMs. Other simple options are changing the LSTMs to other recurrent networks or even a transformer or changing the number of distractor items. Lastly, two model specific changes could be made. For the image model, the ConvNet can be adjusted and for the word embedding model the number of epochs could be increased.

References

Austin, J. L., & Urmson, J. J. O. (1962). *How to do things with words*. Cambridge, Massachusetts: Harvard University Press.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Boldt, B., & Mortensen, D. (2024). A review of the applications of deep learning-based emergent communication.

Bouchacourt, D., & Baroni, M. (2019). Miss tools and mr fruit: Emergent communication in agents learning about object affordances.

Brighton, H., & Kirby, S. (2006). Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, *12*(2), 229-242.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., & Baroni, M. (2020). Compositionality and generalization in emergent languages.

Chaabouni, R., Kharitonov, E., Dupoux, E., & Baroni, M. (2019). Anti-efficient encoding in emergent communication.

Chrabaszcz, P., Loshchilov, I., & Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the cifar datasets.

Cogswell, M., Lu, J., Lee, S., Parikh, D., & Batra, D. (2019). Emergence of compositional language with deep generational transmission.

Deichler, A., Wang, S., Alexanderson, S., & Beskow, J. (2023). Learning to generate pointing gestures in situated embodied conversational agents. *Frontiers in robotics and AI*, *10*, 1110534-1110534.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). Bertje: A dutch bert model.

Erenshteyn, R., Foulds, R., & Galuska, S. (1994). Is designing a neural network application an art or a science? *SIGCHI bulletin*, *26*(3), 23-29.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings.

Evtimova, K., Drozdov, A., Kiela, D., & Cho, K. (2017). Emergent communication in a multi-modal, multi-step referential game.

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning.

Frege, G. (1892). Uber sinn und bedeutung. *Zeitschrift für Philosophie Und Philosophische Kritik*, *100*(1), 25–50.

Fukushima, K. (1969). Visual feature extraction by a multilayered network of analog threshold elements. *IEEE transactions on systems science and cybernetics*, *5*(4), 322-333.

Graesser, L., Cho, K., & Kiela, D. (2019). Emergent linguistic phenomena in multi-agent communication games.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Ingham, J. (1997). What is an agent? In (Vol. 1193, p. 41-43). doi: 10.1007/BFb0013572

Kharitonov, E., Dessì, R., Chaabouni, R., Bouchacourt, D., & Baroni, M. (2021). *EGG: a toolkit for research on Emergence of lanGuage in Games.* https://github.com/facebookresearch/EGG.

Kottur, S., Moura, J. M. F., Lee, S., & Batra, D. (2017). Natural language does not emerge 'naturally' in multi-agent dialog.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations.

Lazaridou, A., & Baroni, M. (2020). Emergent multi-agent communication in the deep learning era.

Lazaridou, A., Hermann, K. M., Tuyls, K., & Clark, S. (2018). Emergence of linguistic communication from referential games with symbolic and pixel input.

Lazaridou, A., Peysakhovich, A., & Baroni, M. (2017). Multi-agent cooperation and the emergence of (natural) language.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature (London)*, *521*(7553), 436-444.

Lewis, D. D. K. (1969). *Convention : a philosophical study.* Cambridge, Mass: Harvard University Press.

Li, F., & Bowling, M. (2019). Ease-of-teaching and language structure from emergent communication.

Li, Y., Ponti, E. M., Vulić, I., & Korhonen, A. (2020). Emergent communication pretraining for few-shot machine translation.

Piantadosi, S. T., Tily, H., & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, *122*(3), 280-291.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in bertology: What

we know about how bert works. *Transactions of the Association for Computational Linguistics*, *8*, 842-866.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386-408.

Skyrms, B. (2010). *Signals: evolution, learning, & information.* Oxford: Oxford University Press.

Steels, L. (2003). Social language learning. *The future of learning*, 133–162.

Szandała, T. (2020). Review and comparison of commonly used activation functions for deep neural networks.

Tenney, I., Das, D., & Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., ... Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need.

Vogt, P. (2005). The emergence of compositional structures in perceptually grounded language games. *Artificial intelligence*, *167*(1), 206-242.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229-256.

Wittgenstein, L. (1953). *Philosophical investigations.* Blackwell, Oxford, UK. (Translated by G.E.M. Anscombe)

Zubek, J., Korbak, T., & Rączaszek-Leonardi, J. (2023). Models of symbol emergence in communication: a conceptual review and a guide for avoiding local minima.