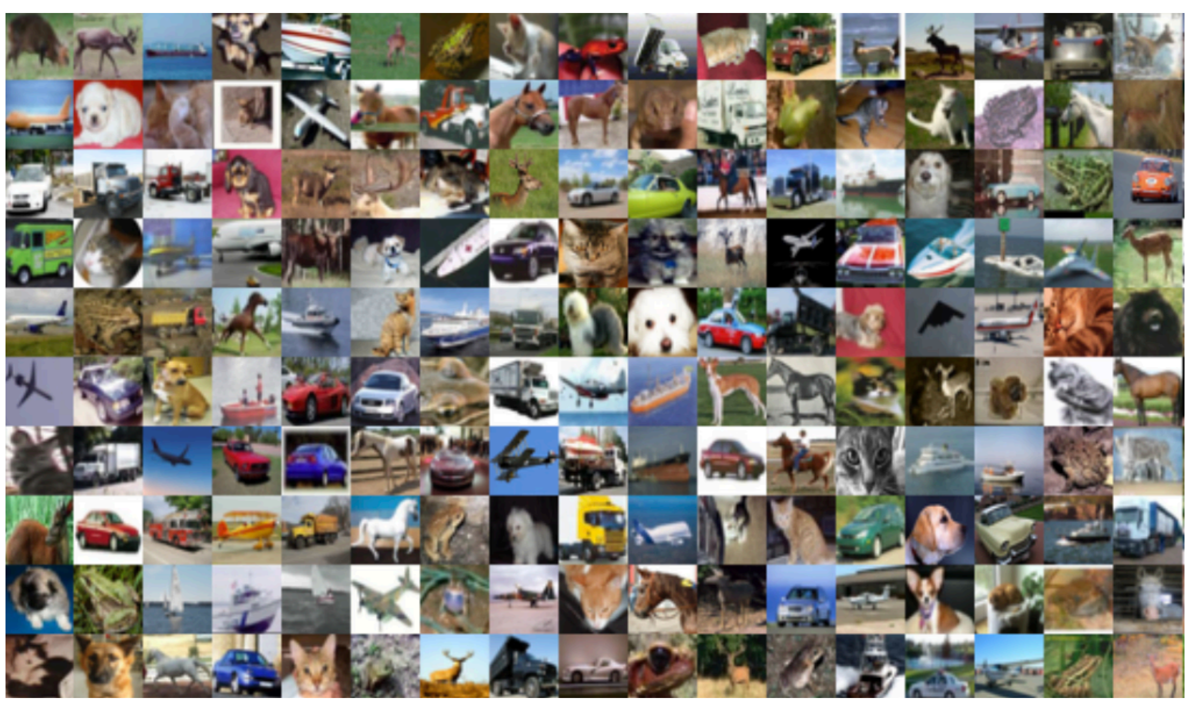
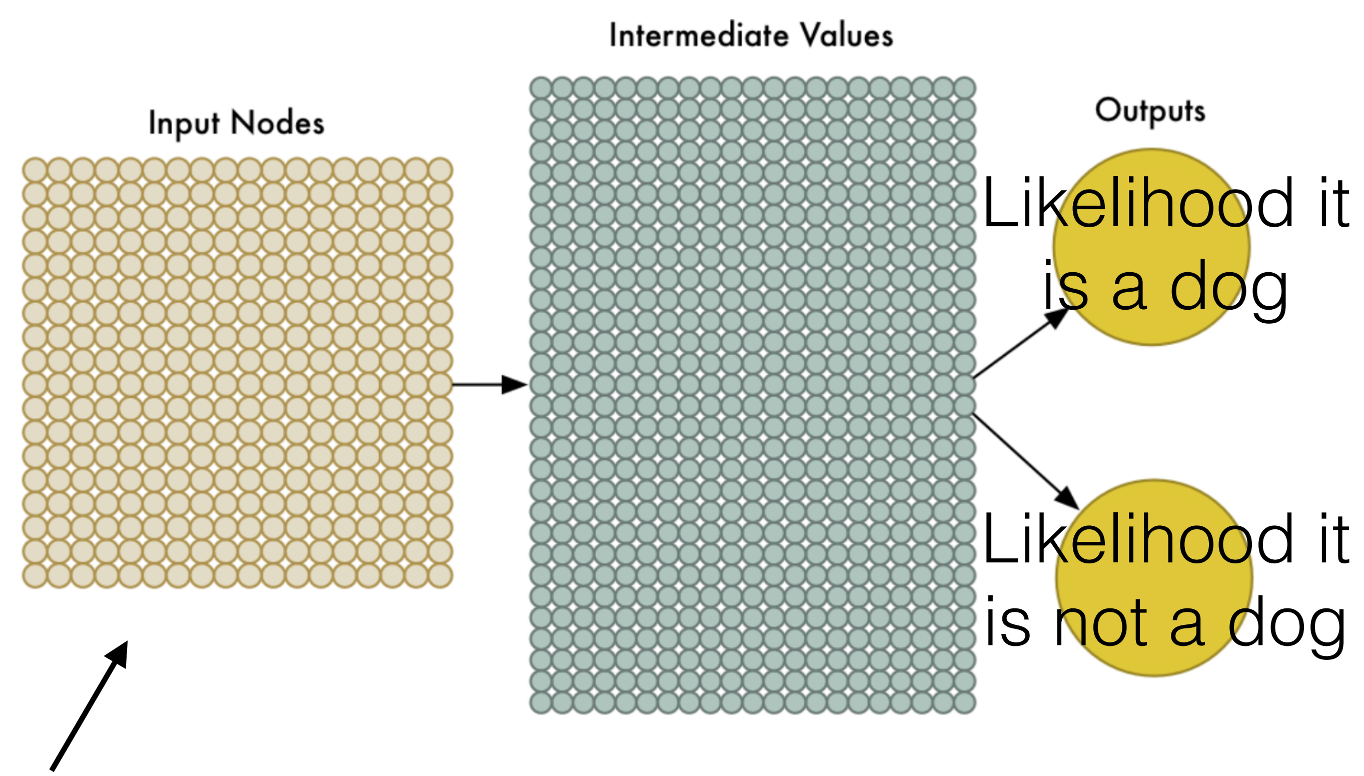
Machine Learning - Part 4

Apr. 17, 2025

Recap question:

You are building a neural network to be able to classify dogs in a large dataset of images. What would you use as training data and what labels would you use?

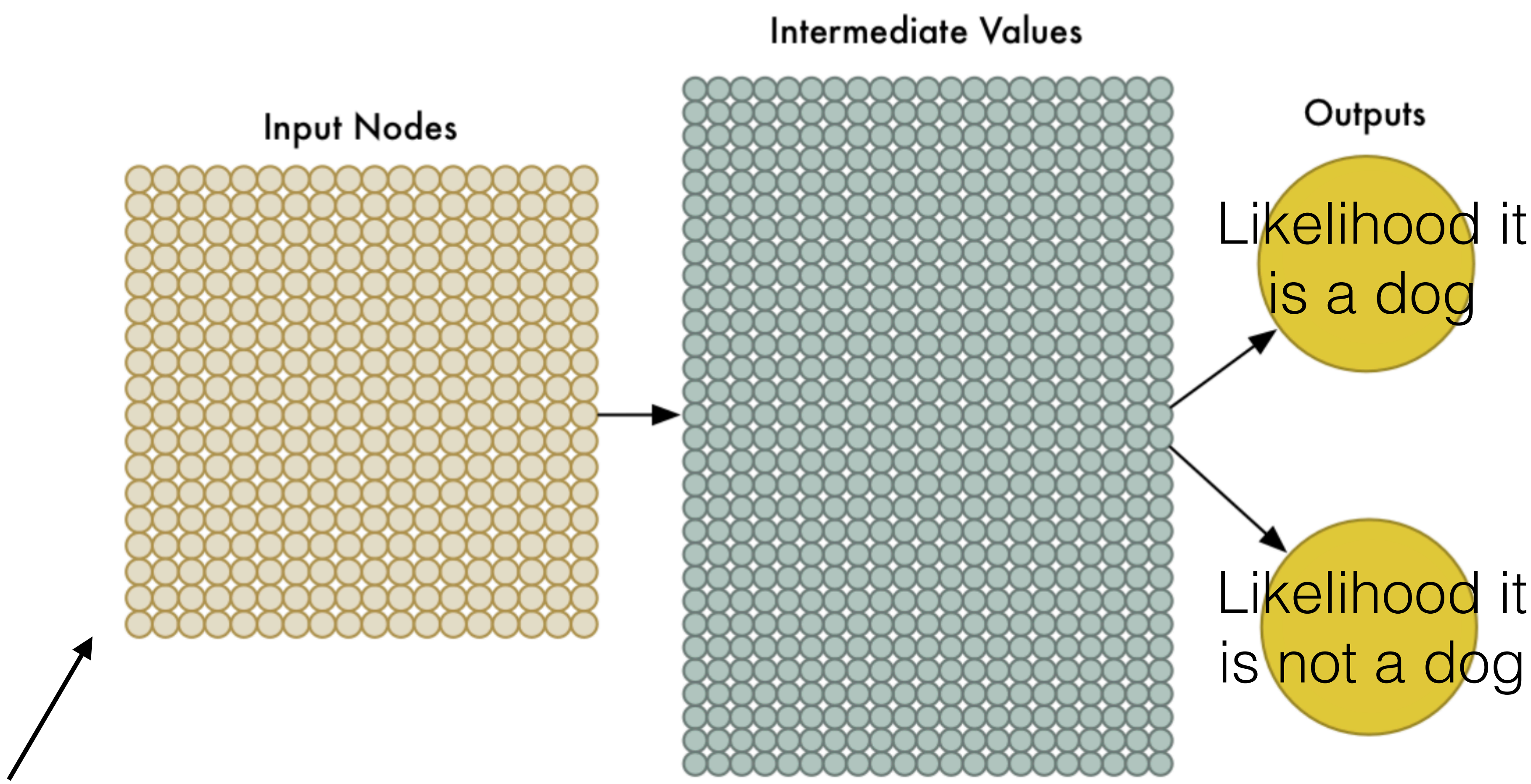




As many neurons
as pixels in image

Recap question:

For training data, we would you use both pictures of dogs (labelled as 100% being dogs) and pictures of all the other categories in the dataset (labelled as 100% **NOT** being dogs)



As many neurons as pixels in image

Machine Learning - Part 4

April 17, 2025

By the end of this lecture, you will be able to:

1. Explain how to use neural networks to translate text into other languages
2. Define a recurrent neural network and encoder-decoders
3. Discuss the pro and cons of large language model

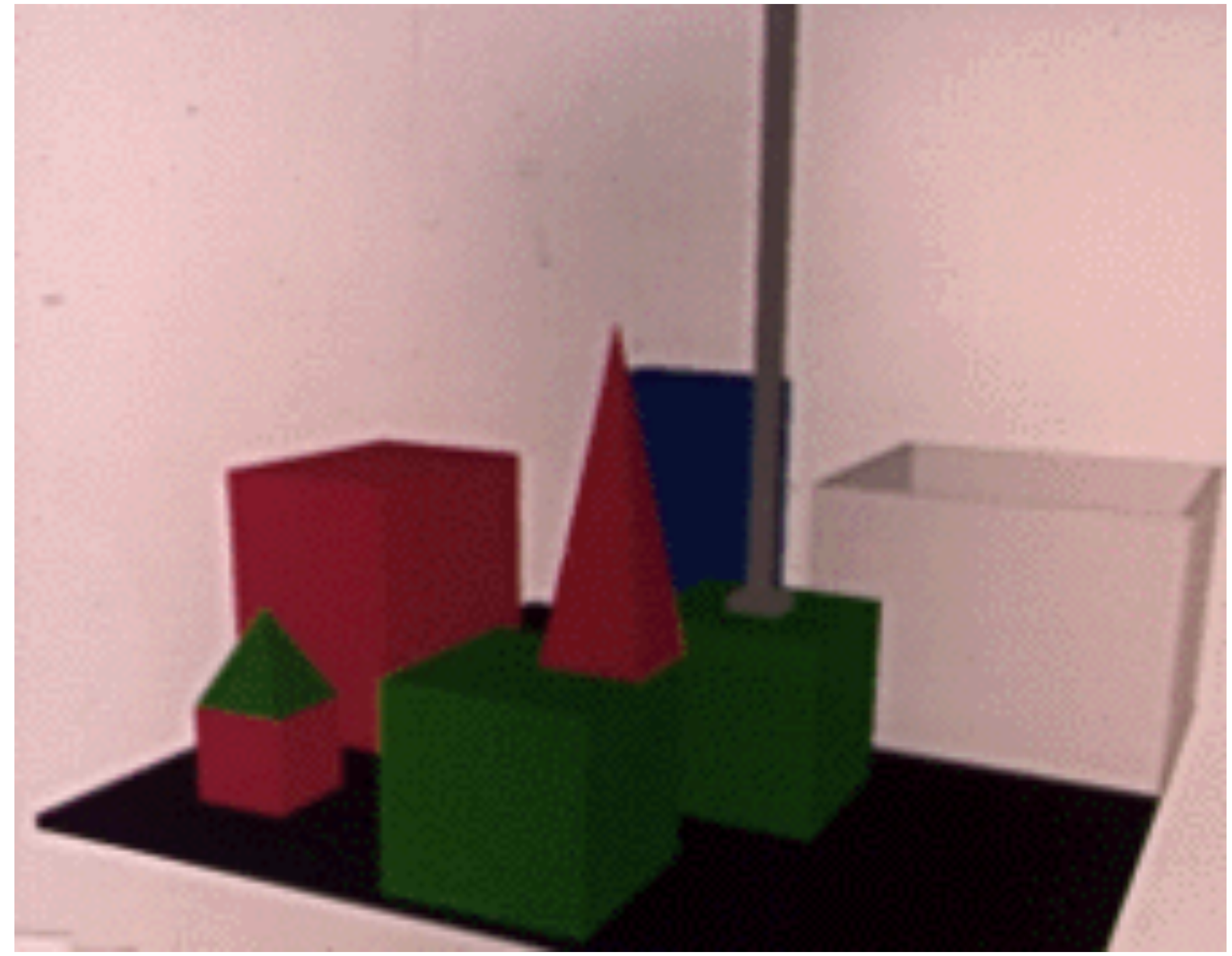
Natural Language Processing (NLP)

Natural language processing (**NLP**) is the field concerned with how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents or instructions.

Rule-based techniques, 1960s - 1990s

Make the computer recognize a list of keywords and tell it what to respond when it sees the keywords

Example: SHRDLU, 1968



Person: Pick up a big red block.

Computer: OK.

Person: Grasp the pyramid.

Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.

Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.

Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

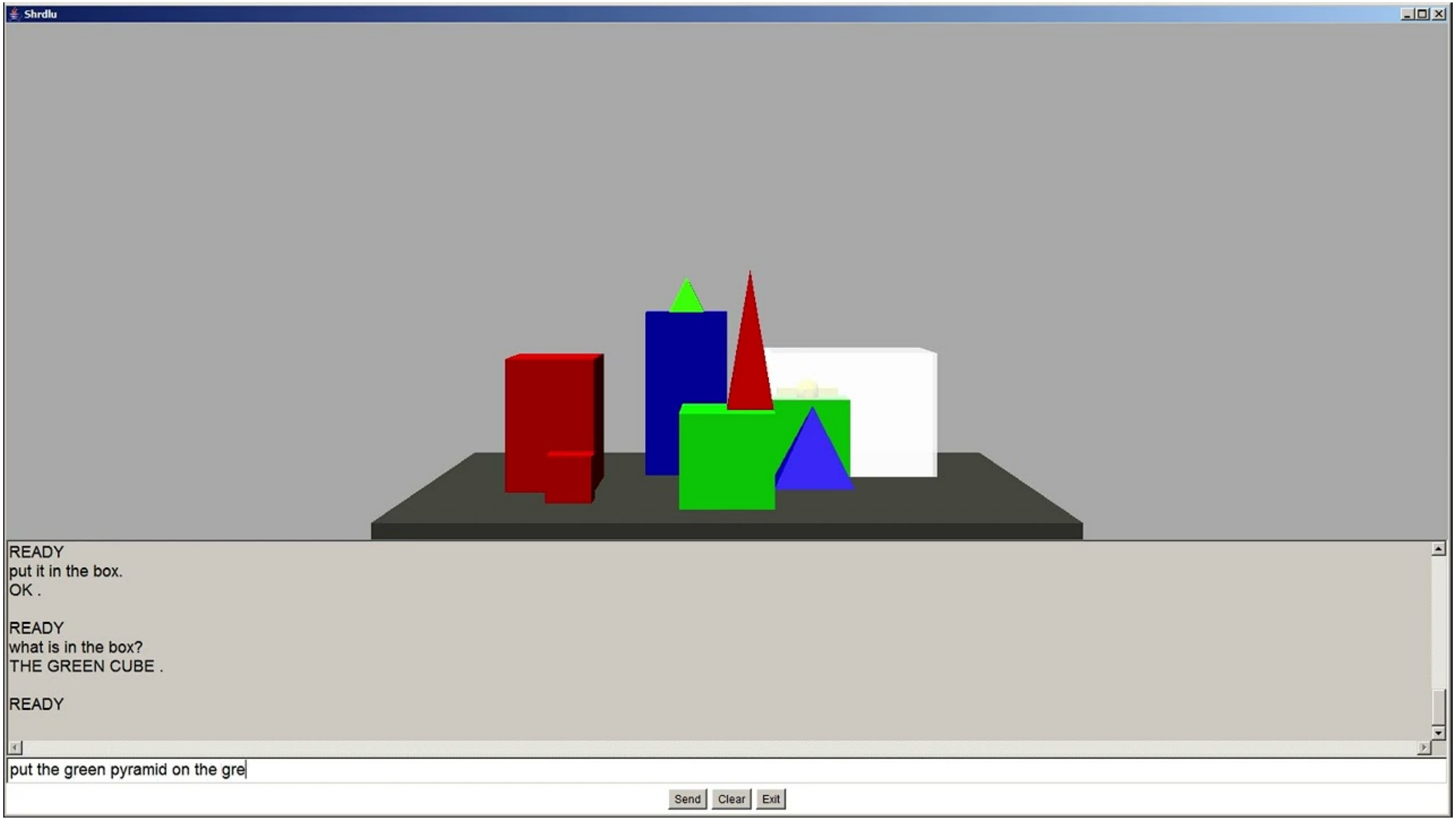
Computer: OK.

Person: What does the box contain?

Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.

Person: What is the pyramid supported by?

Computer: THE BOX.

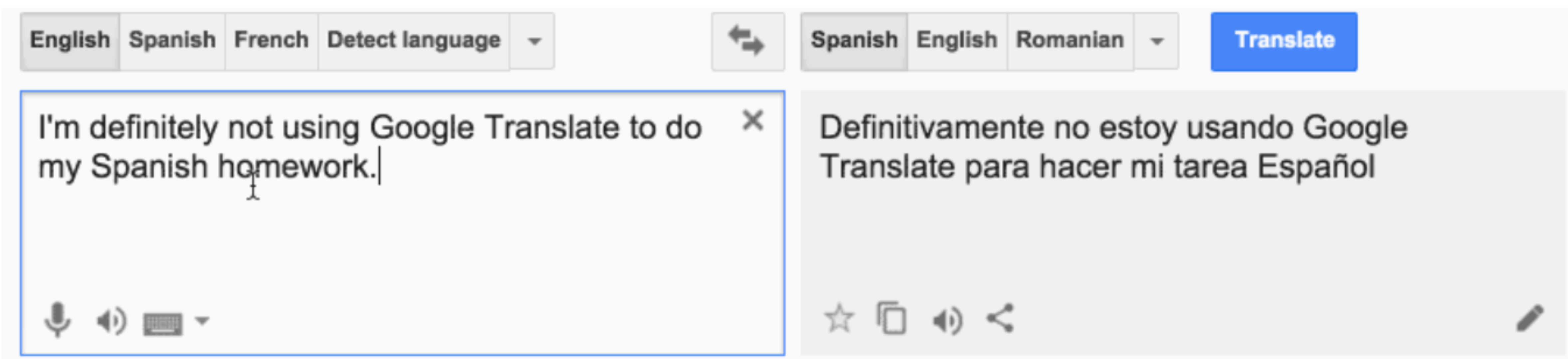

Shrdlu
READY
put it in the box.
OK .
READY
what is in the box?
THE GREEN CUBE .
READY
put the green pyramid on the gre
Send
Clear
Exit

Example: PARRY, 1972

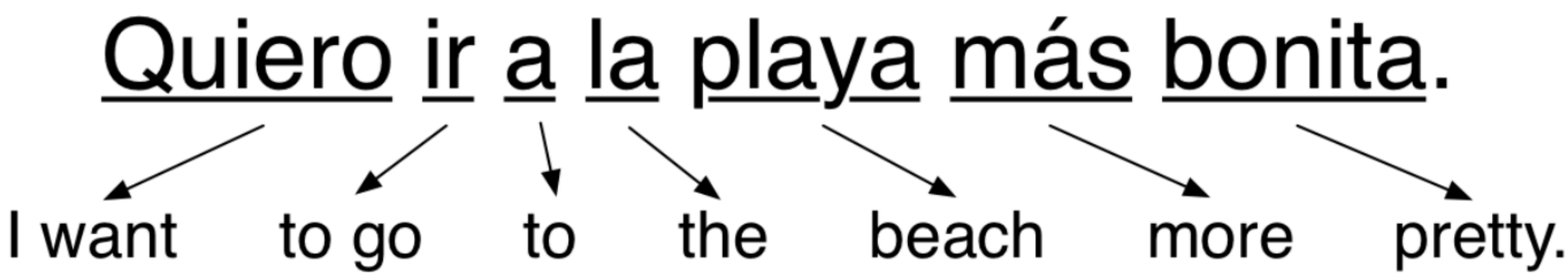
```
PARRY — Good Evening.

    ELIZA — Tell me your problems. Please terminate input with
            a period or a question mark.

People get on my nerves sometimes.

    I am not sure I understand you fully.

You should pay more attention.

    Suppose you should pay more attention.
```

Psychiatrists can only guess correctly between computer and human 48% of the time for PARRY.

Example: Machine translation

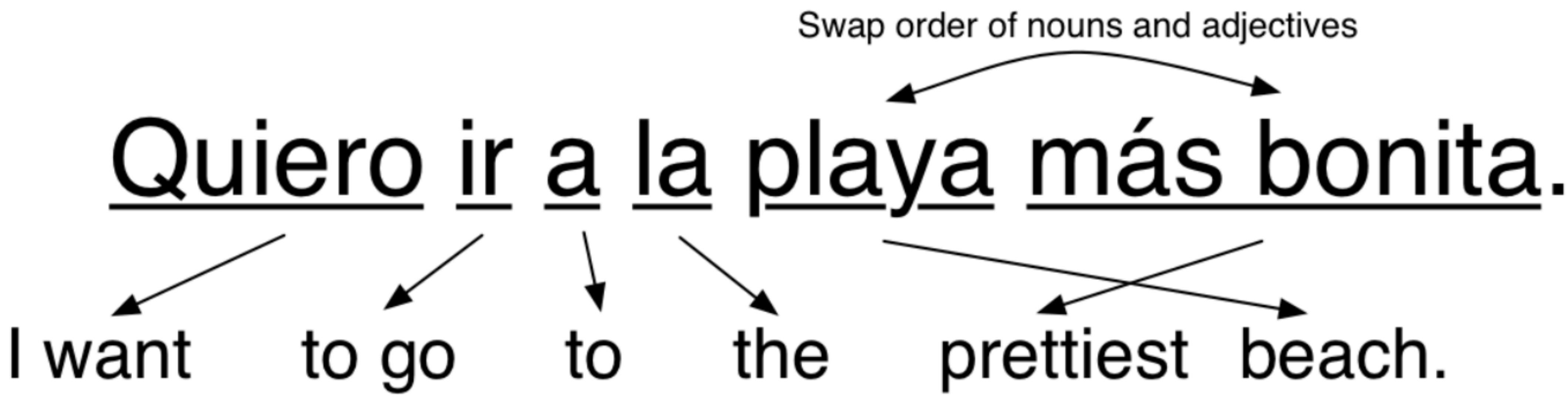


Making computers translate



We just replace each Spanish word with the matching English word.

Replace every word in a sentence with the translated word in the target language—easy to implement but the results are bad



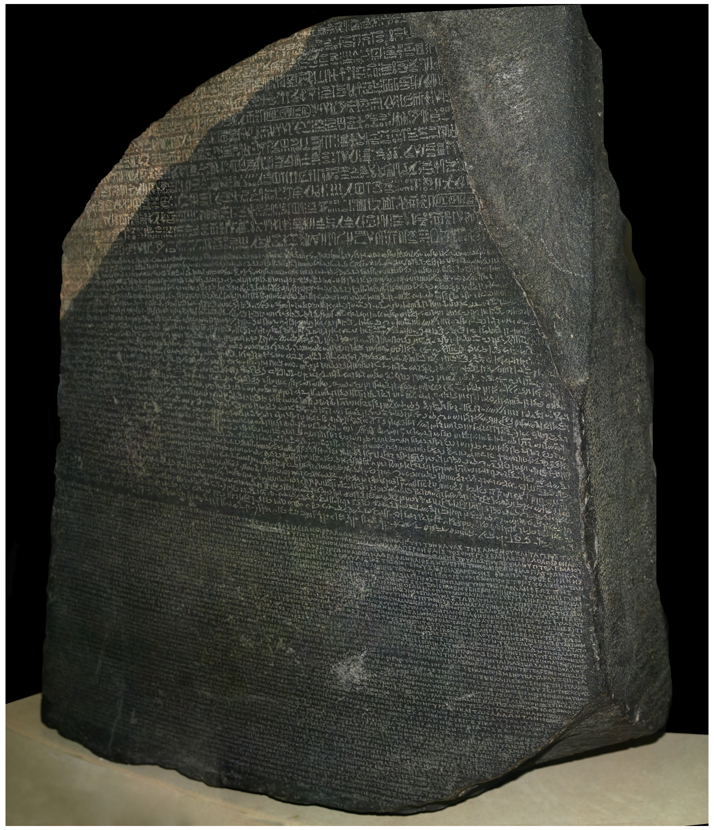
1. translate common two-word phrases as a single group
2. swap the order nouns and adjectives since they usually appear in reverse order in Spanish from how they appear in English.

The problem is that human language **doesn't follow a fixed set of rules**. Human languages are full of special cases, regional variations, and just flat out rule-breaking. The way we speak English is more influenced by who invaded who hundreds of years ago than it is by someone sitting down and defining grammar rules.

Statistical techniques, 1990s - 2010s

After the failure of rule-based systems, new translation approaches were developed using models based on probability and statistics instead of grammar rules.

Building a statistics-based translation system requires lots of training data where the exact same text is translated into at least two languages. This double-translated text is called ***parallel corpora***.



In the same way that
the Rosetta Stone was used
by scientists in the 1800s to
figure out Egyptian
hieroglyphs from Greek,
computers can use parallel
corpora to guess how to
convert text from one
language to another.

The fundamental difference with statistical translation systems is that they don’t try to generate one exact translation. Instead, they generate **thousands of possible translations** and then **rank** those translations by how likely each is to be correct. They estimate how “correct” something is by how similar it is to the training data.

Step 1: Break original sentence into chunks

<u>Quiero</u> <u>ir</u> <u>a</u> <u>la playa</u> <u>más bonita</u>.

Step 2: Find all possible translations for each chunk

Even the most common phrases have lots of possible translations.

We are seeing how actual people translated these same chunks of words in real-world sentences. This helps us capture all of the different ways they can be used in different contexts.

Step 3: Generate all possible sentences and find the most likely one

Just from the chunk translations we listed in Step 2, we can already generate nearly 2,500 different variations of our sentence by combining the chunks in different ways.

> *I love | to leave | at | the seaside | more tidy.*
> *I mean | to be on | to | the open space | most lovely.*
> *I like | to be |on | per the seaside | more lovely.*
> *I mean | to go | to | the open space | most tidy.*

In a real-world system, there will be even more possible chunk combinations because we'll also try different orderings of words and different ways of chunking the sentence.

> *I try | to run | at | the prettiest | open space.*
> *I want | to run | per | the more tidy | open space.*
> *I mean | to forget | at | the tidiest | beach.*
> *I try | to go | per | the more tidy | seaside.*

Now we need to scan through all of these generated sentences to find the one that sounds “most human.”

To do this, we compare each generated sentence to millions of real sentences from books and news stories written in English. The more English text we can get our hands on, the better.

> *I try | to leave | per | the most lovely | open space.*

The sentence would not be very similar to any sentences in our data set, and we'll give this possible translation a low probability score.

> *I want | to go | to | the prettiest | beach.*

This sentence will be similar to something in our training set, so it will get a high probability score.

Pro: Statistical machine translation systems perform much better than rule-based systems if you give them enough training data.

Con: They are complicated to build and maintain. Every new pair of languages you want to translate requires experts to tweak and tune a new multi-step translation pipeline.

Because it is so much work to build these different pipelines, trade-offs have to be made. In the old days, if you asked Google to translate Georgian to Telugu, it has to internally translate it into English as an intermediate step because there’s not enough Georgian-to-Telugu translations happening to justify investing heavily in that language pair.

Because it is so much work to build these different pipelines, trade-offs have to be made. In the old days, if you asked Google to translate Georgian to Telugu, it has to internally translate it into English as an intermediate step because there's not enough Georgian-to-Telugu translations happening to justify investing heavily in that language pair. And it might do that translation using a less advanced translation pipeline than if you had asked it for the more common choice of French-to-English.

Ultimate goal?

The ultimate goal of machine translation is a black box system that learns how to translate by itself—just by looking at training data. With statistical machine translation, humans are still needed to build and tweak the multi-step statistical models.

Neural network techniques, 2010s -

In 2014, researchers found a way to apply deep learning to build this black box system. Their deep learning model takes in a *parallel corpora* and uses it to learn how to translate between those two languages without any human intervention.

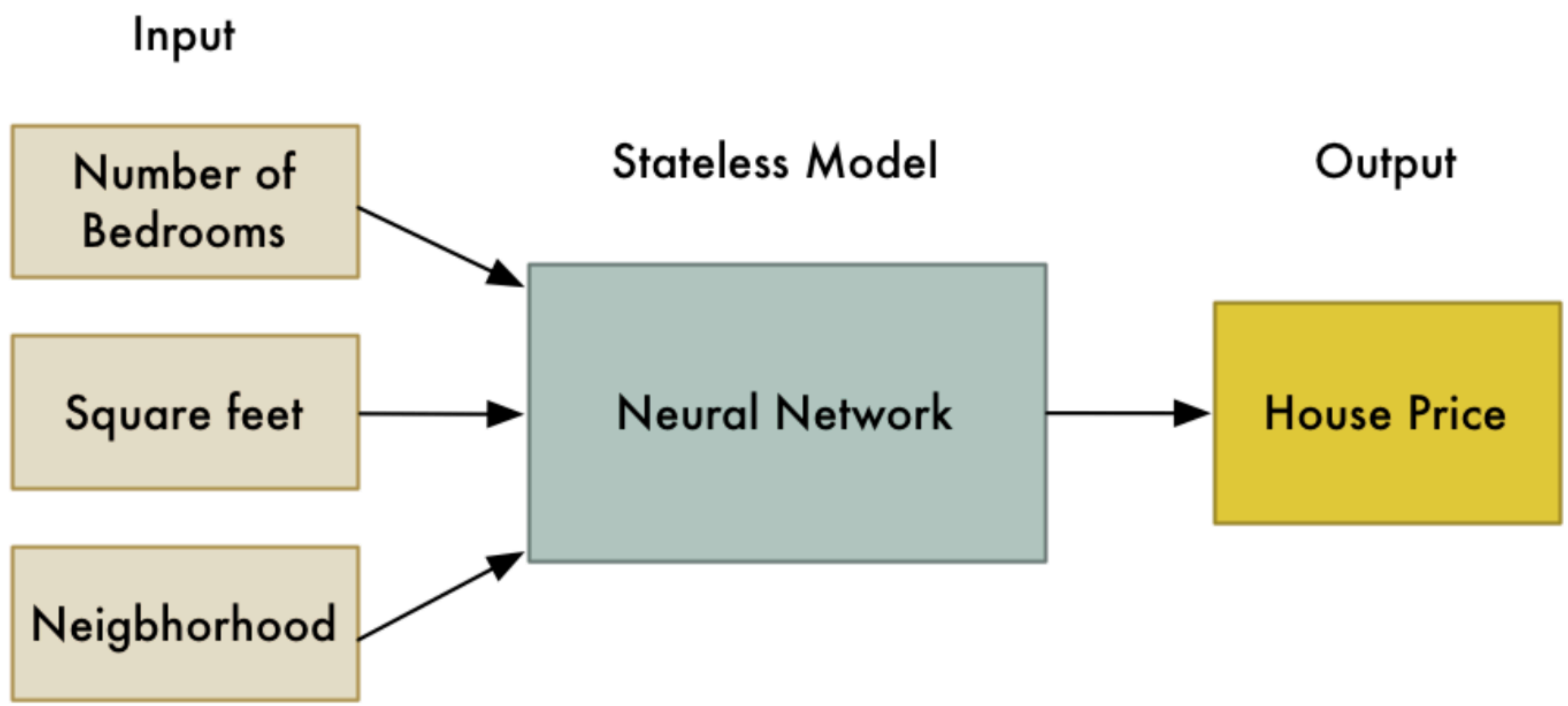
This is made possible by combining two ideas—*recurrent neural networks* and *encodings*.

Make things even simpler

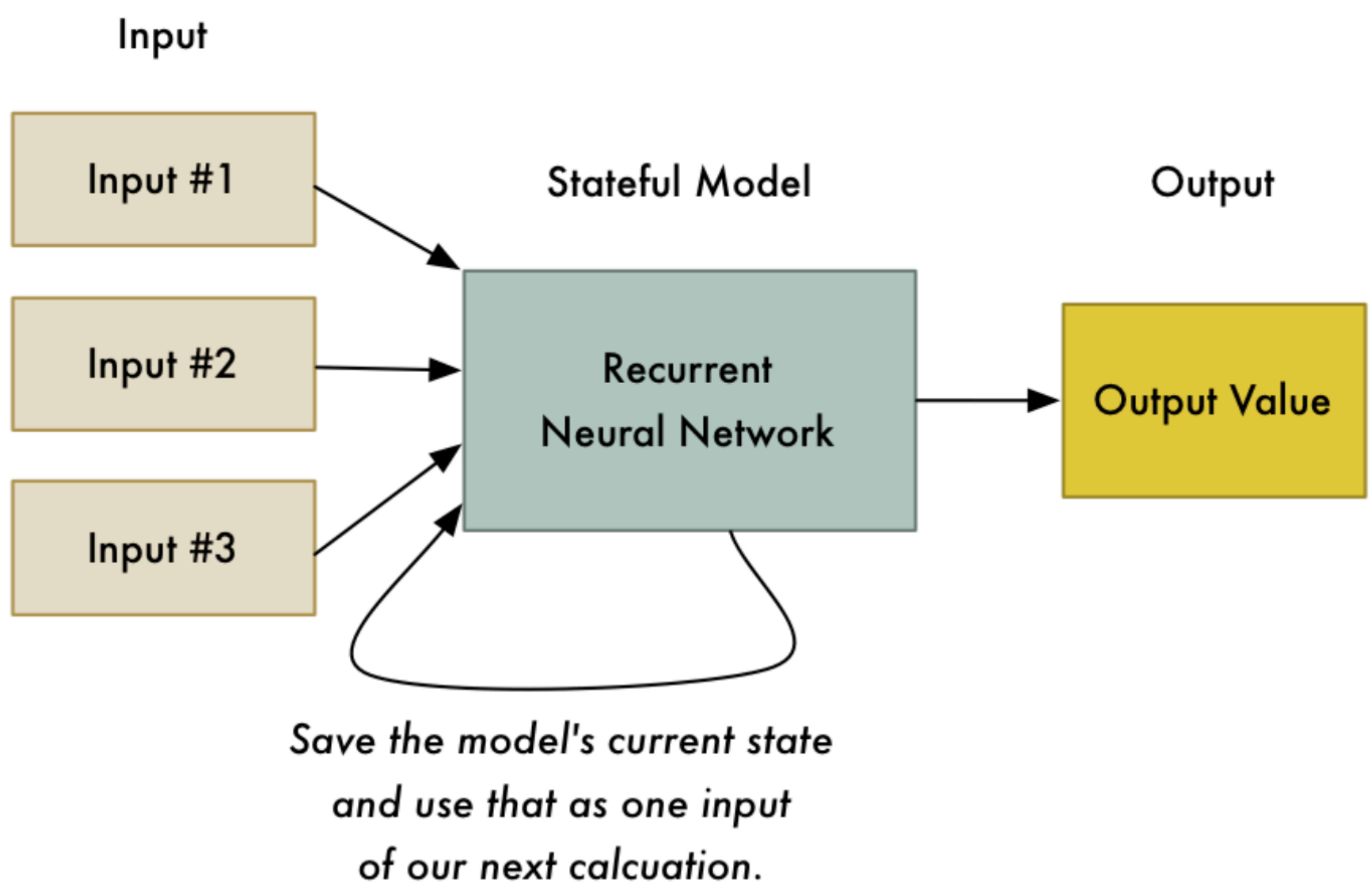
Input: Machine learning is fun

Output: Machine learning is fun

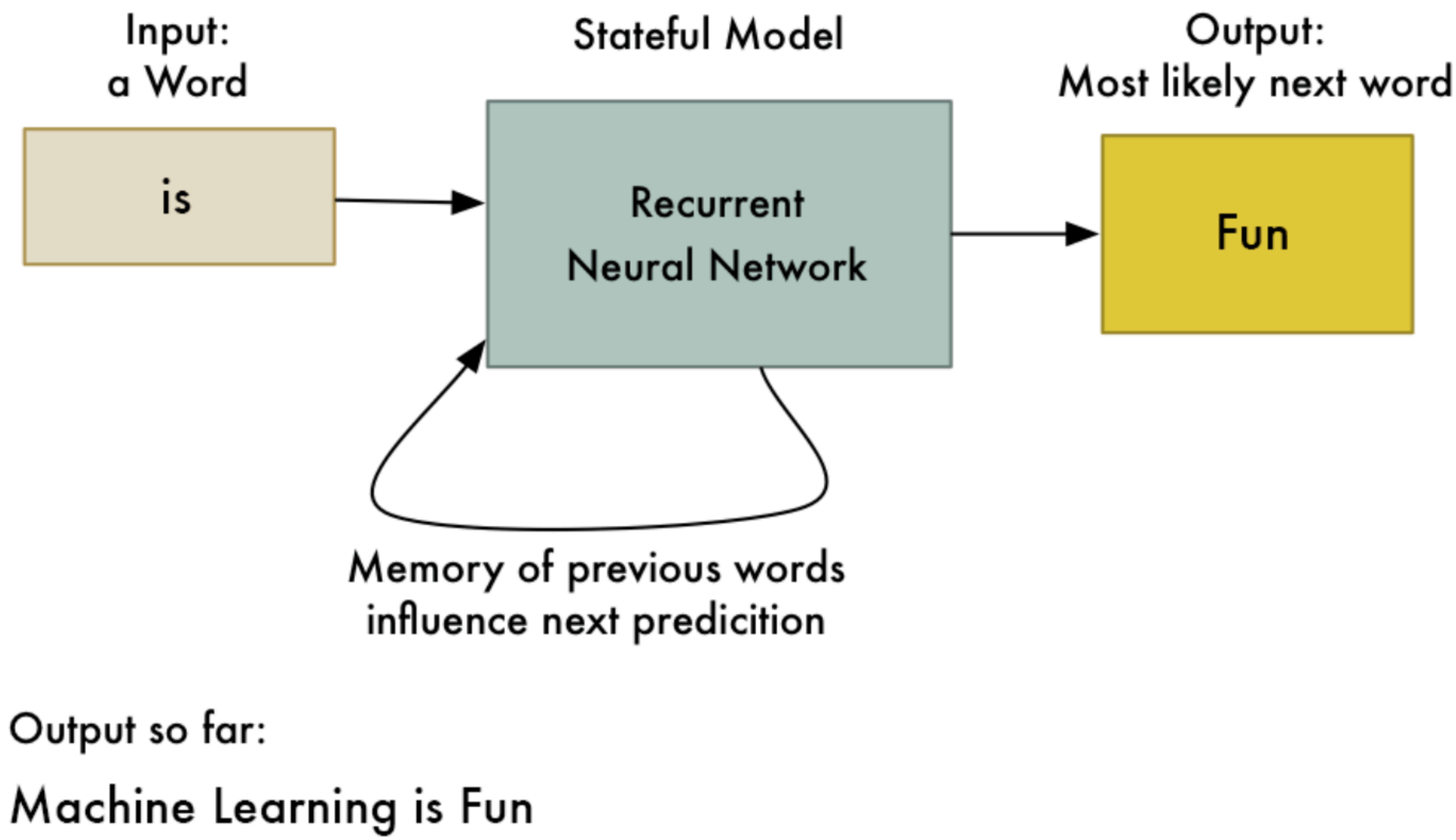
Just like **cryptography!**



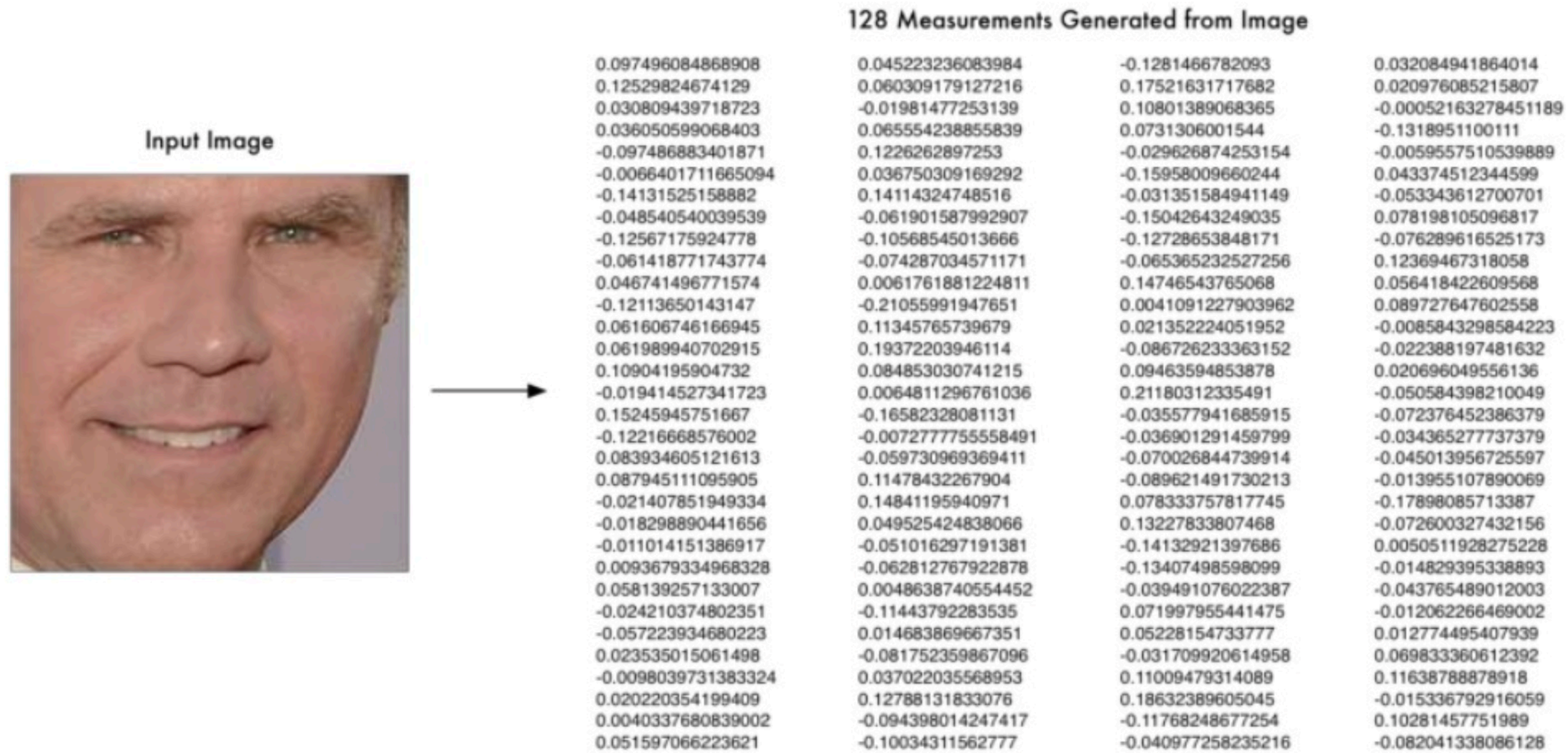
A neural network takes in a list of numbers and calculates a result. Like most machine learning algorithms, these neural networks are *stateless.* That is to say, they have no memory of past calculations.



In a recurrent neural network, the previous state of the neural network is one of the inputs to the next calculation. This means that current calculations change the results of future calculations!

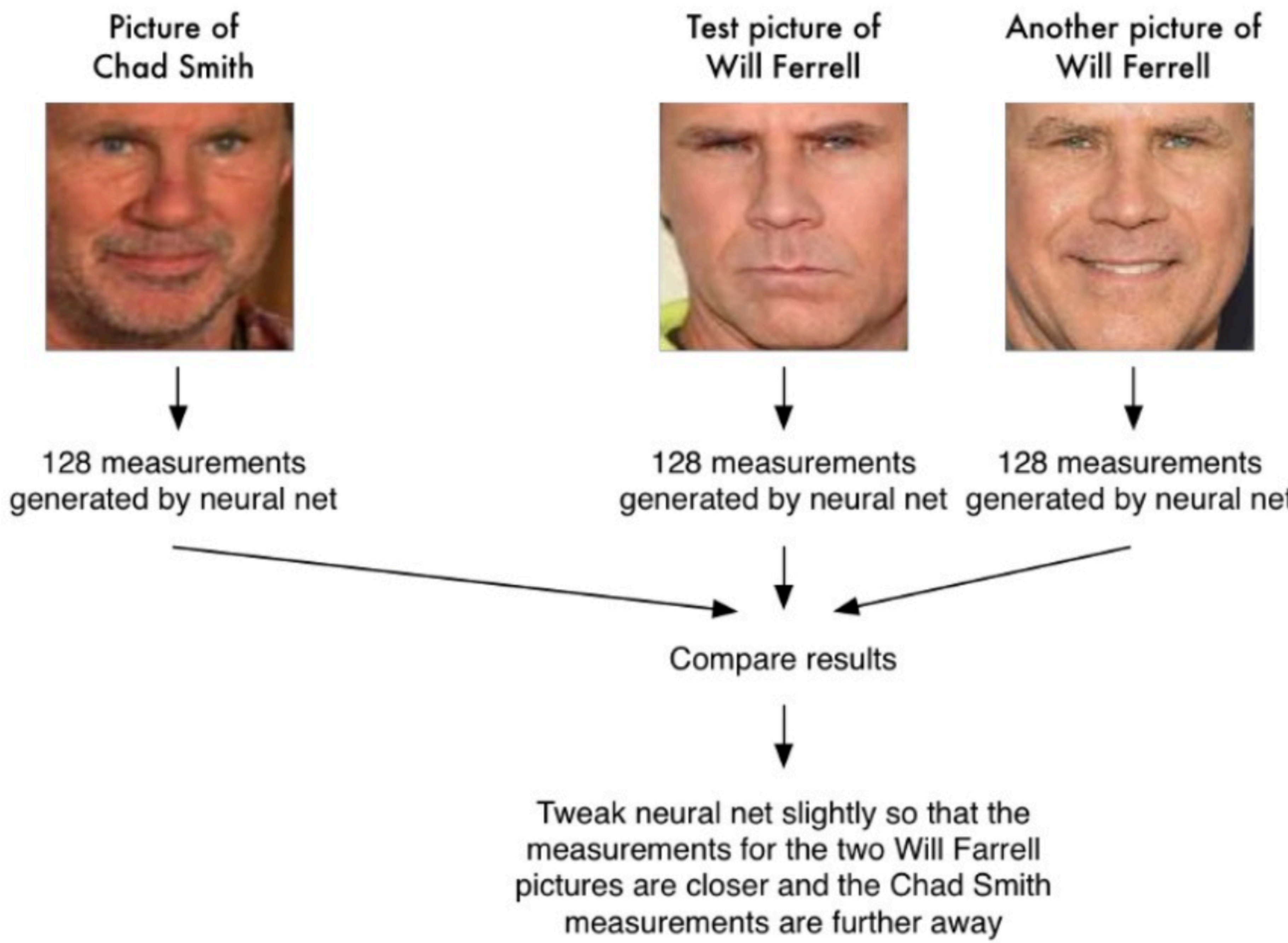


This trick allows neural networks to learn patterns in a sequence of data. For example, you can use it to predict the next most likely word in a sentence based on the first few words, or translate spoken text word for word instead of waiting until the person is done speaking.



The idea of turning a face into a list of measurements is an example of an *encoding*. We are taking raw data (a picture of a face) and turning it into a list of measurements that represent it (the encoding).

A single “triplet” training step



It lets us represent something very complicated (a picture of a face) with something simple (128 numbers). Now comparing two different faces is much easier because we only have to compare these 128 numbers for each face instead of comparing full images.

We can do the same thing with sentences by coming up with an encoding that represents every possible different sentence as a series of unique numbers.

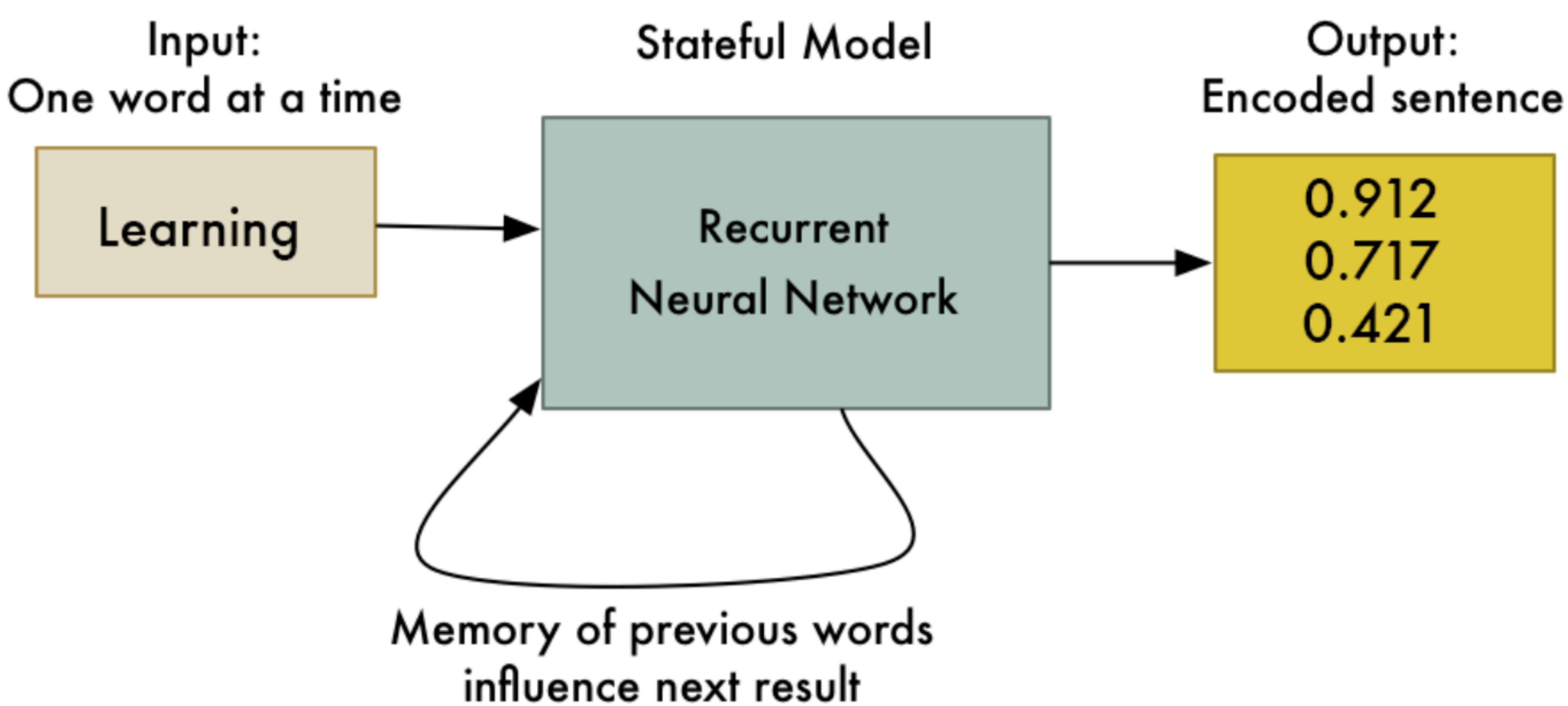
Input Sentence

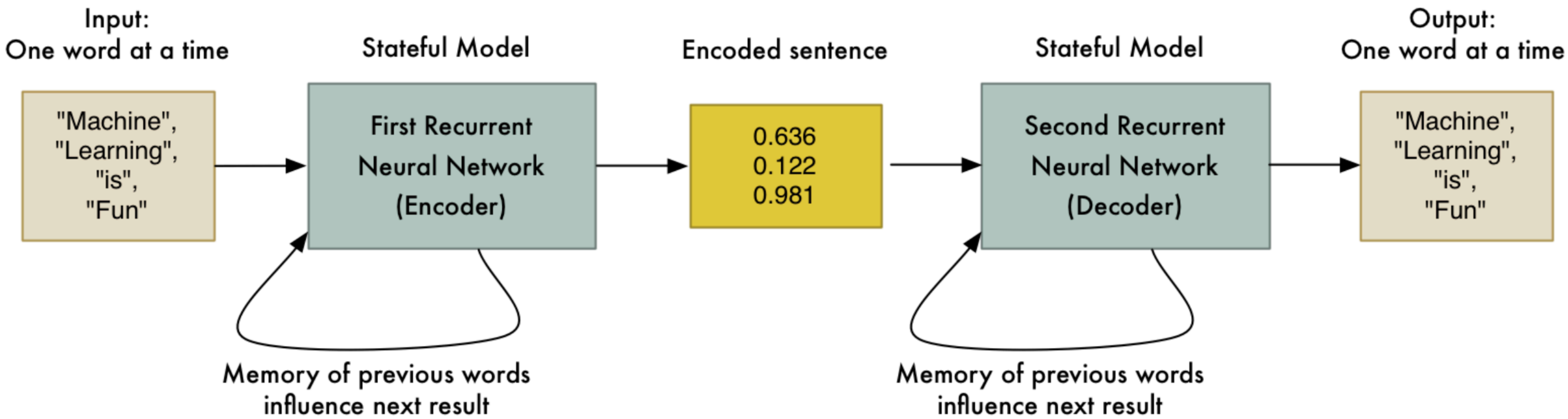
"Machine Learning is Fun!" →

Measurements Generated from Sentence

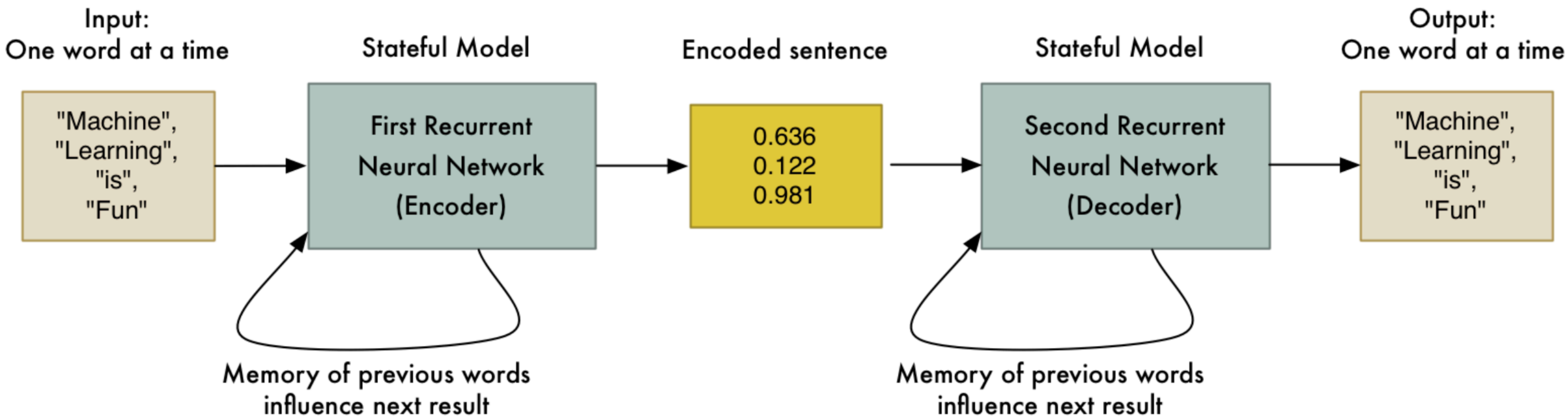
0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

To generate this encoding, we'll feed the sentence into the RNN, one word at a time. The final result after the last word is processed will be the values that represent the entire sentence.

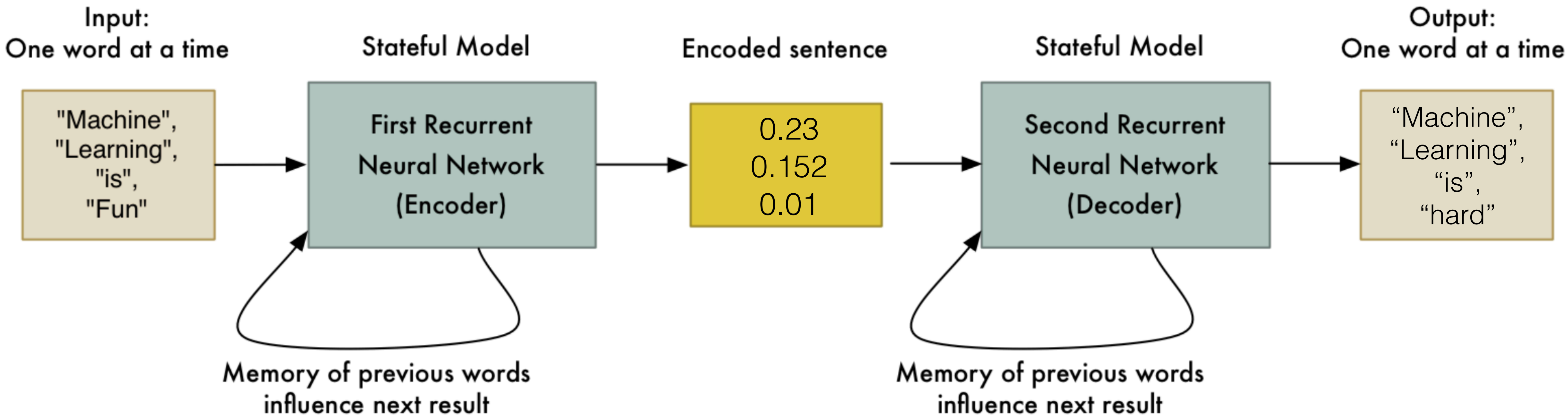




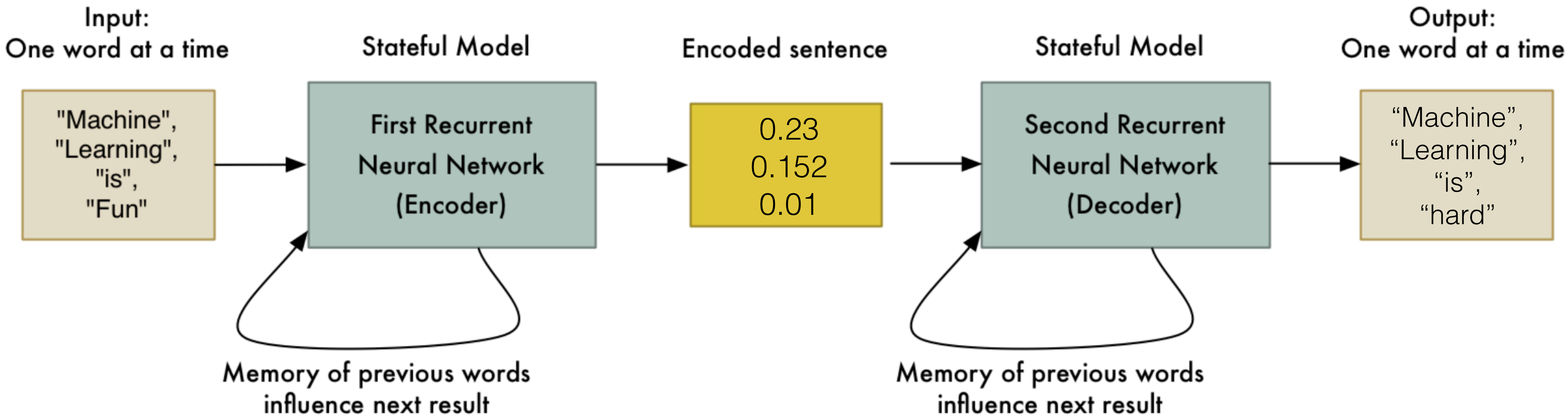
We can connect two RNNs end-to-end. The first RNN can generate the encoding that represents a sentence. Then the second RNN can take that encoding and just do the same logic in reverse to decode the original sentence again.



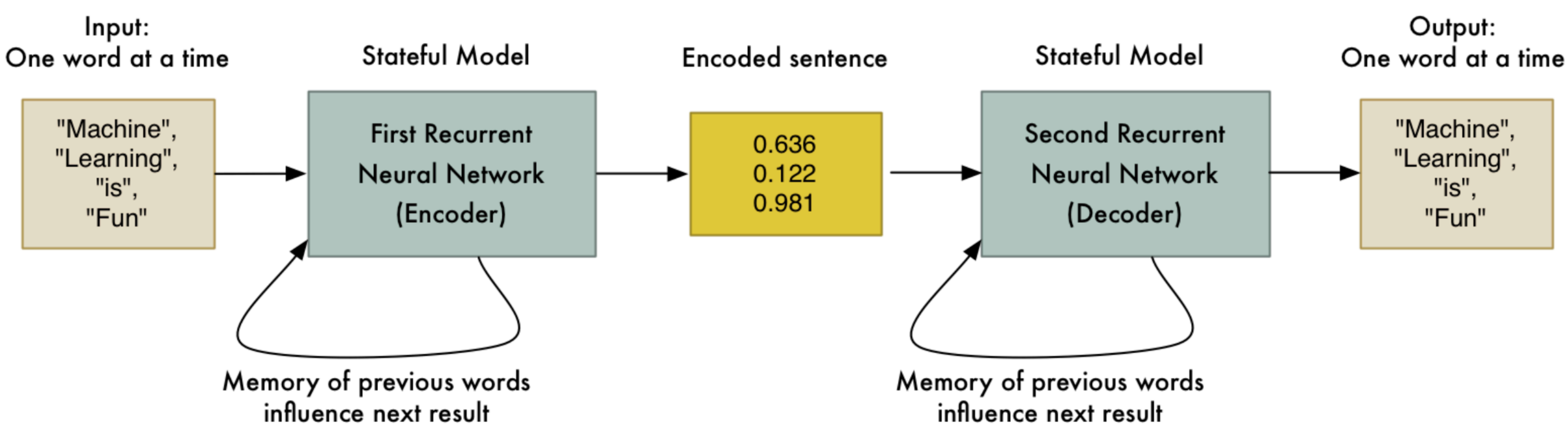
We train the encoder-decoder pair by ensuring that the output of the neural network is equal to the input. We then let the values of the neurons in the hidden layer be the encoded sentence.

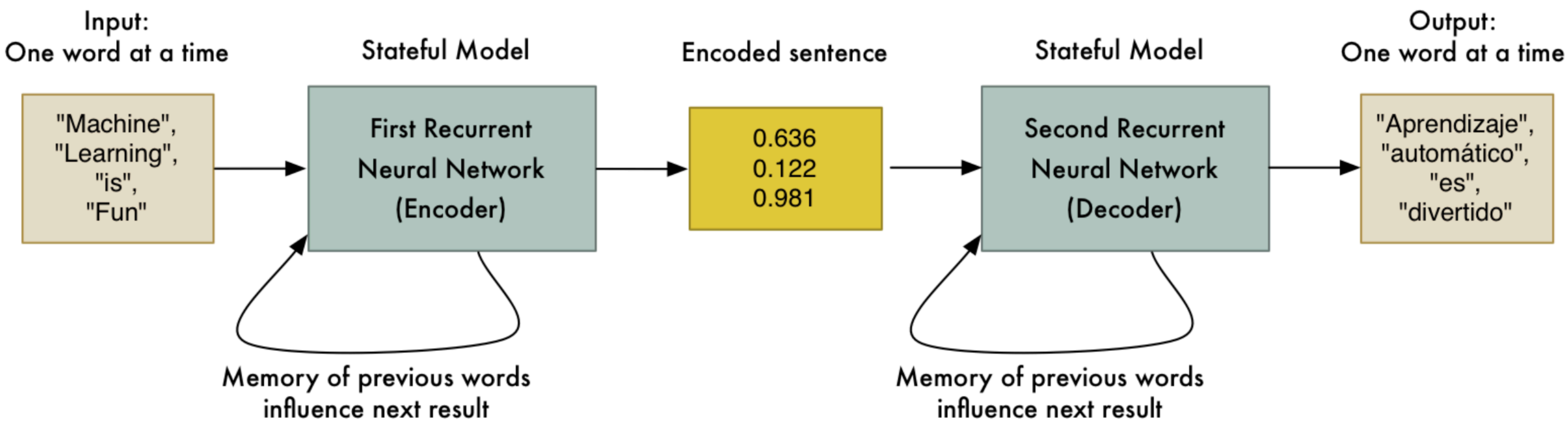


Since the output is not equal to the input, the weights chosen in the recurrent networks above are wrong, so we keep trying new weights, until



Since the output is not equal to the input, the weights chosen in the recurrent networks above are wrong, so we keep trying new weights, until



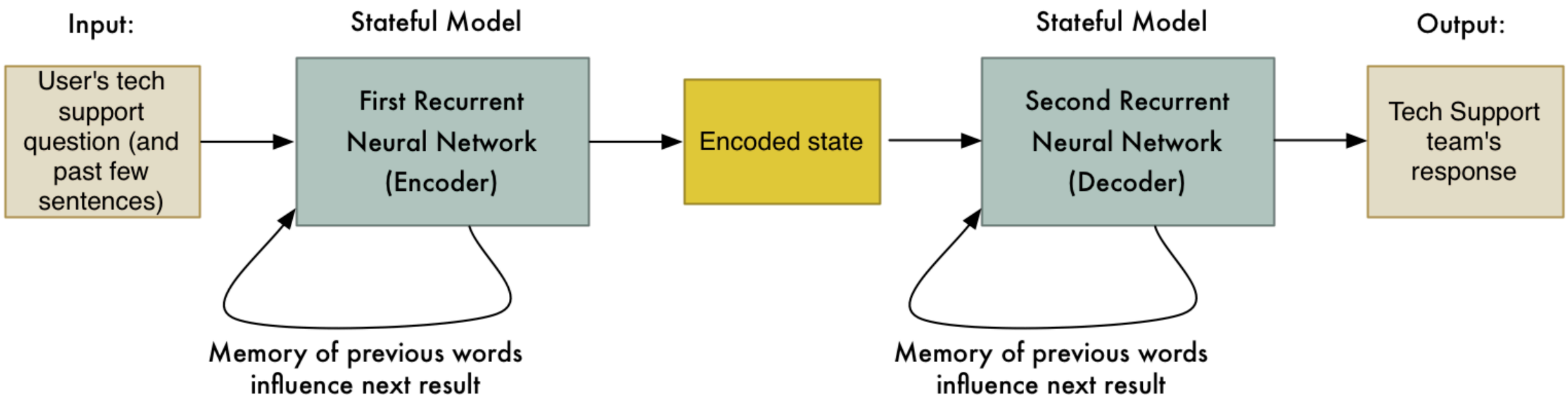


What if we train the encoder-decoder pair to decode the sentence into Spanish instead of English? We could use our *parallel corpora* training data to train it to do that

This framework doesn’t depend on knowing any rules about human language. **The algorithm figures out those rules itself**. This means you don’t need experts to tune every step of your translation pipeline. The computer does that for you.

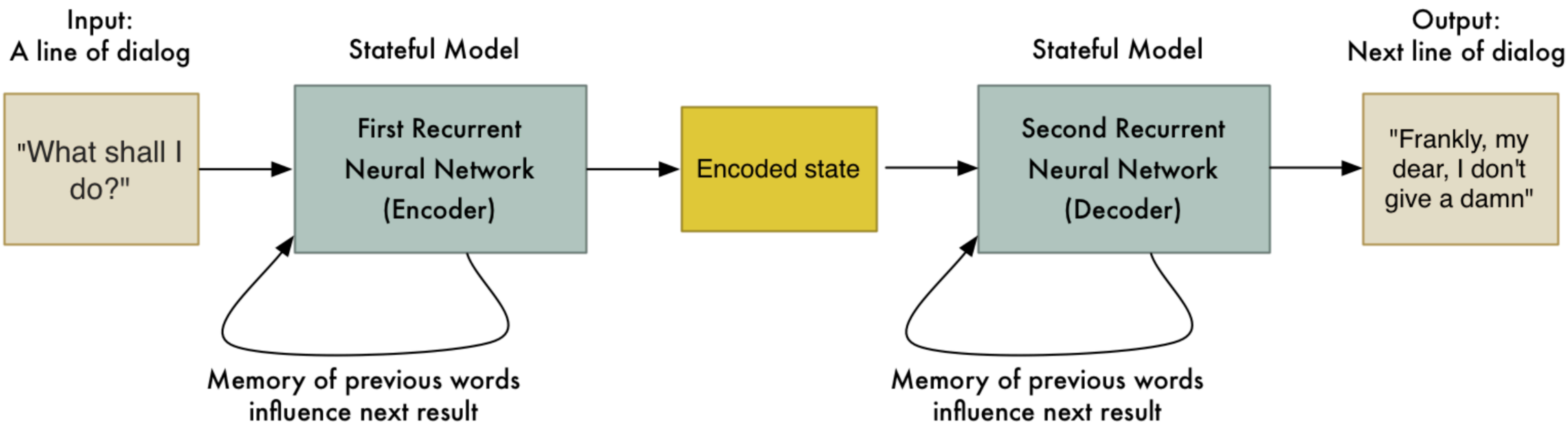
Moreover, this approach works for almost any kind of sequence-to-sequence problem.

AI chat bot

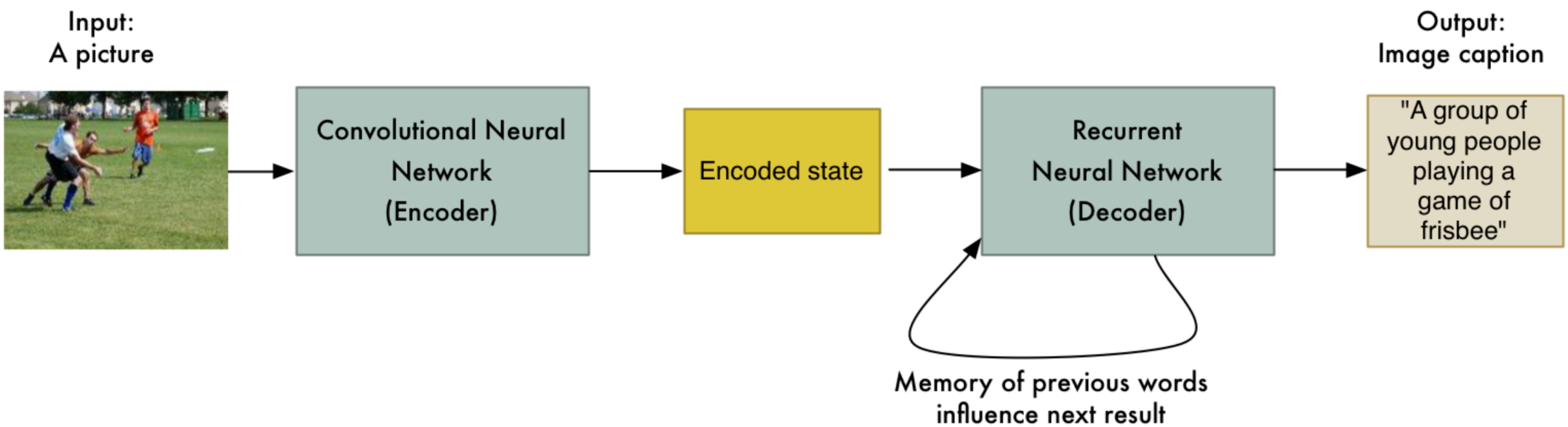


The chat logs between employees and the tech support team were captured at Google. Then a sequence-to-sequence model was trained where the employee's question was the input sentence and the tech support team's response was the "translation" of that sentence.

When a user interacted with the bot, they would "translate" each of the user's messages with this system to get the bot's response. The end result was a semi-intelligent bot that could (sometimes) answer real tech support questions.

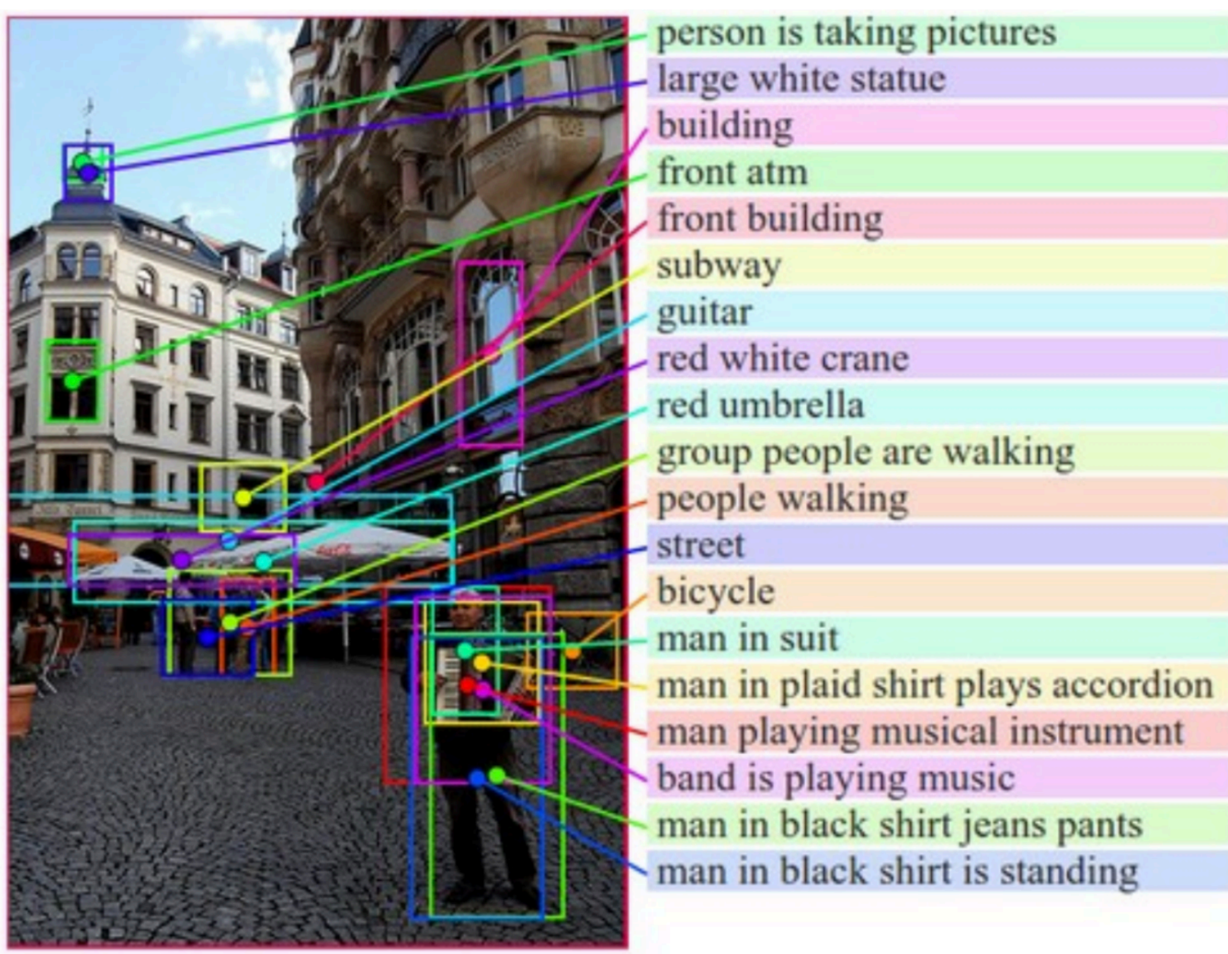


They also tried building a chat bot based on millions of movie subtitles. The idea was to use conversations between movie characters as a way to train a bot to talk like a human. The input sentence is a line of dialogue said by one character and the “translation” is what the next character said in response.



Replacing the first RNN with a convolutional neural network allows the input to be a picture instead of a sentence. We can then turn pictures into words or words into pictures (as long as we have lots of training data).

The idea can be used to build a system capable of describing images in great detail by processing multiple regions of an image separately.



This makes it possible to build image search engines that are capable of finding images that match oddly specific search queries.

Revolution in NLP

A new Neural Network building block, the **transformer,** was discovered in 2017. It is essentially a more powerful replacement of RNNs and encoder-decoders.

It is the basis for ChatGPT and similar tools. GPT stands for Generative Pre-trained Transformer.

Basically, transformer is more powerful in encoding.

ChatGPT

Developer: OpenAI

Initial release: Nov. 30 2022

Engine: GPT-3.5 and GPT-4 (Paid)

Other competing programs: Copilot, Claude,
Jasper, BERT, …

How can I help you today?

Give me ideas
for what to do with my kids' art

Recommend a dish
to bring to a potluck

Dive into history
choose a historical figure

Tell me a fun fact
about the Golden State Warriors

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Brief History of OpenAI

https://en.wikipedia.org/wiki/OpenAI

2015-2018: Non-profit beginnings

In Dec 2015, Sam Altman, Greg Brockman, Reid Hoffman, Jessica Livingston, Peter Thiel, Elon Musk, Amazon Web Services (AWS), Infosys, and YC Research announced the formation of OpenAI and pledged over $1 billion to the venture.

2019: Transition from non-profit

In 2019, OpenAI transitioned from non-profit to "capped" for-profit, with the profit being capped at 100 times any investment.

2020-2023: ChatGPT

In 2020, OpenAI announce GPT-3, a language model trained on large internet datasets.

Brief History of OpenAI

https://en.wikipedia.org/wiki/OpenAI

2024–present: Public/non-profit efforts, Sora

On February 15, 2024, OpenAI announced a text-to-video model named Sora, which it plans to release to the public at an unspecified date

https://www.youtube.com/watch?v=HK6y8DAPN_0

Discussion

What can be done pretty well by ChatGPT now?

1. Who is …? What is …?
2. What will be the next step for …?
3. Write me a paragraph about …?
4. What does the following sentence mean?
5. Any more idea?

Discussion

What can be done pretty well by ChatGPT now?

1. Answering questions
2. Offering suggestions
3. Generating text and creative writing
4. Language translation
5. …

Discussion

What can possibly be done by future large language models?

Discussion

What can possibly be done by future large language models?

1. Improved understanding
2. Enhanced personalization
3. Emotional intelligence
4. Expert-level knowledge
5. Incorporate other modalities

Discussion

What can not be done by ChatGPT even in the long run?

Discussion

What can not be done by ChatGPT even in the long run?

1. Critical thinking and reasoning

2. Original thought

3. Sensitive information

4. ….