Self-Aware LSTM-Based Agents

By Ryan Mukai

WORD COUNT 6341
TIME SUBMITTED 08-JUN-2022 08:49PM
PAPER ID 87029630

Self-Aware LSTM-Based Agents

Introduction

Goal: A Simple Form of Self-Awareness

The goal of this work is the creation of neural-network-based agents that can, given a knowledge base of Boolean logic sentences and a question regarding the value of a Boolean variable, tell us whether the variable in question is ~~T~~true, ~~F~~false, or unknown. ~~-~~In addition to solving simple Boolean logic problems, these agents should exhibit self-awareness and be capable of exploiting this feature ~~self-awareness~~ to solve a problem cooperatively. Because ~~Since~~ the term "self-awareness" can be defined in different ways~~ly~~, we provide our definition ~~of self-awareness~~ below~~:~~.

1. An agent must know whether it possesses adequate knowledge to solve a problem.
    a. If it possesses adequate knowledge, it should solve the problem.
    b. If it lacks adequate knowledge, it should request help solving the problem.
2. An agent must know whether its internal knowledge base is contradictory because a contradictory knowledge base in Boolean logic permits any conclusion to be drawn.
    a. An agent with a contradictory state of knowledge should report ~~state~~ this instead of trying to provide an answer because ~~since~~ delivering a solution is impossible in this case.
3. An agent should be capable of providing the contents of its knowledge base upon request.
    a. This implies that the ~~an~~ agent has explicit knowledge of its knowledge base contents, which is another form of self-knowledge.

For the purposes of this~~is~~ work, agents fulfilling ~~capable of performing~~ these three criteria~~tasks above~~ are considered "self-aware." ~~-~~Self-awareness allows agents with contradictory knowledge states to warn the user and avoid providing a wrong answer. ~~-~~It also ~~will~~ enable~~s~~ agents to cooperate. ~~-~~For example, suppose that an agent lacks adequate knowledge to solve a problem and requests help from a second agent. ~~-~~The second agent's knowledge may be sufficient to allow the first agent to solve the problem. ~~-~~Here, agent self-awareness permits~~enables~~ cooperation because ~~since~~ the first agent knows that it must request help (by being aware of the inadequacy of its knowledge state). The second agent has direct, symbolic knowledge of the contents of its knowledge base and can provide those contents on demand.

In this work, we create neural~~-~~-network-based agents capable of solving simple problems in Boolean logic. ~~-~~Neural networks for solving Boolean logic problems and other problems in symbolic mathematics are not new. ~~-~~Using neural networks for logical entailment, which is very closely connected ~~related~~ to such ~~this~~ problem~~s~~, ~~i~~was discussed in (Evans, Saxton, Amos, Pushmeet, & Grefenstette, 2018). Moreover, the ~~closely~~ related issue of using neural networks for symbolic mathematics was~~is~~ covered in (Lample & Charton, 2019). ~~-~~The present work differs from these two reports because it emphasizes self-awareness and cooperation in neural-network-based agents, a topic not covered in the cited ~~two~~ works

Commented [A1]: The manuscript was edited according to the conventions of U.S. English. Because no target journal had been indicated during manuscript upload, no specific style or formatting points were applied during editing except where necessary for internal consistency. Please ensure that all required manuscript elements are added prior to submission, e.g., author names, affiliations, contact details, an abstract, conflicts of interest, acknowledgments, and so on.

It would be worth considering whether the 15 sections of the manuscript (excluding the "References" section) would be better organized as subsections under the standard headings of "Introduction", "Methods" (or "Theory" or similar), and "Results and Discussion" etc. In this editor's opinion, this is perhaps the main issue with the manuscript at present, at least in the context of a typical journal article – each of the 15 sections appeared to read as a mixture of background information, theory, and results and discussio ...

Commented [A2]: It is generally preferable to avoid abbreviations in titles, although depending on the target publication this may be acceptable here, assuming that the typical reader can be expected to be familiar with the meaning (the expanded version "Self-Aware Long Short- ...

Commented [A3]: It was assumed that this section has yet to be written (as opposed to being a first-level heading containing the "Goal: A Simple Form of Self-Awareness" section as a second-level heading). It did seem that much of the text in the latter section could reasonably be included ...

Commented [A4]: It is suggested to use lower case consistently here (with the exception of the quoted answers "TRUE" and "FALSE" later in the manuscript). Please confirm this and similar edits.

Commented [A5]: Please confirm that the edited text has retained the intended meaning (note that the above three points, as currently written, are not "tasks" per se).

Commented [A6]: A mixture of single and double quotation marks had been used throughout the manuscript, with periods/commas sometimes inside the quotation marks and sometimes outside. For consistency, these were changed to double quotation marks (standard in U.S. ...

Field Code Changed

Commented [A7]: Please confirm this edit or otherwise clarify, as the intended meaning of "this problem" (singular) was unclear – no problem has been explicitly stated, and it was assumed that this refers to the problems (plural) mentioned in the previous two sentences.

Commented [A8]: As the in-text citations are still present as field codes from the reference management software, they were not directly edited. However, please note that the current style/sentence structure around these citations is incorrect – when author names are mentioned explicitly ...

Field Code Changed

~~cited above~~. Self-aware neural ~~-~~network systems ~~were~~are surveyed in (Du, et al., 2020), where ~~and~~ the type of self-awareness ~~covered in this survey~~ focuse~~d~~s on the implementation and operation~~al~~ of the neural network itself, with the network having awareness of its own execution environment and ~~with~~ the ability to optimize its dataflow, resource use ~~of resources~~, and execution within that environment for performance optimization. This is a very different type of self-awareness than what is described here, because ~~since~~ this paper focuses on awareness of one's state of knowledge, not on awareness of neural network execution environment and performance.

The present work is organized as follows. We start by defining the propositional (Boolean) logic problems that our agents are trained to solve and then explain the behavior of the logical agents that we have created. Next, we provide an overview of agent architecture, consisting of a knowledge base (implemented as a Python list of strings) and an LSTM neural network. The basic aspect~~s~~s of LSTM neural networks and the particular encoder–decoder architecture used in the present work are then described.

Propositional Logic Problems

Our treatment of propositional logic very closely follows that used in (Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, 2021), with our notation, along with many direct quotations, from (Norvig, aima-python, n.d.). ~~Specifically~~ ~~In particular~~, our notation is summarized in Table 1 ~~described as follows:~~

Operation	Book	Python Infix Input	Python Output	Python `Expr` Input
Negation	¬ P	~P	~P	`Expr('~', P)`
And	P ∧ Q	`P & Q`	`P & Q`	`Expr('&', P, Q)`
Or	P ∨ Q	`P | Q`	`P | Q`	`Expr('|', P, Q)`
Inequality (Xor)	P ≠ Q	`P ^ Q`	`P ^ Q`	`Expr('^', P, Q)`
Implication	P ⟶ Q	`P | '==>' | Q`	`P ==> Q`	`Expr('==>', P, Q)`
Reverse Implication	Q ⟵ P	`Q | '<==' | P`	`Q <== P`	`Expr('<==', Q, P)`
Equivalence	P ⟷ Q	`P | '<=>' | Q`	`P <=> Q`	`Expr('<=>', P, Q)`

Table ~~1~~1: Logical Notation

~~Throughout this paper, t~~The notation given ~~used~~ in the "Python Output" column of Table 1 ~~Table 1: Logical Notation~~ will be used throughout this paper, as is done in our demonstration system. The ~~P~~propositional logic variables are denoted by capital letters "A" through "J" (inclusive).

Field Code Changed

Commented [A9]: As per the earlier comment, it is suggested to consider organizing the manuscript under the standard headings. However, it also appears that some of the final sections ("Data Sets for Training and Testing" onward) are not adequately described here – it is suggested to provide a brief description of all of the sections present. Please check.

Commented [A10]: Would "Propositional Logic Notation" perhaps be a more suitable heading for this section? Please check.

Field Code Changed

Field Code Changed

Commented [A11]: Following on from the previous comment, the in-text citation style will depend on the target journal, but in general book titles and edition numbers should not be included in author–year citations. If it is necessary to mention the title, this could be written as "...very closely follows that used in *Artificial Intelligence: A Modern Approach* (Norvig, 2021)".

For the GitHub citation, it is suggested to check the target journal for any preferred style, but it seemed that neither "aima-python" nor "n.d." should be present (should the latter be a year?). Some guidance about citing GitHub repositories can be found here and here, but in the absence of any guidelines by the target journal it would seem reasonable to follow the standard author–year format and provide the full details in the reference list, e.g., as an onl...

Commented [A12]: Please confirm this and similar edits. Figures and tables should generally be cited explicitly in the t, rather than using "as follows" or similar.

Field Code Changed

Formatted: Do not check spelling or grammar

Commented [A13]: In this and subsequent tables/figures, it is suggested to capitalize only the first word of each entry/text label and proper nouns etc., as is standard, e.g. ...

Commented [A14]: Because no target journal had been indicated, the table and figure captions were edited only for internal consistency. However, it is suggested to check the ...

Commented [A15]: Please note that it is not necessary to repeat table/figure captions in cases such as this – the ...

Field Code Changed

Commented [A16]: The intended meaning of "as is done in our demonstration system" could not be understood – which system is being referred to here? Please clarify.

Commented [A17]: Some of the letters defined here do not seem to be used in the manuscript itself – is more information needed here? Please check.

Logical Agents

Agents in our system are trained to perform propositional inference. ~~—~~An agent is presented with a set of sentences and ~~is~~ then asked about the truth value of a propositional variable. For example, ~~For example,~~ a simple case of modus ponens would be as follows:

Input ~~Sentences~~sentences	A A ==> B
Question	What is B ?
Answer	B is true.

Agents are based on an LSTM sequence-to-sequence neural network as described in the following two ~~s~~Sections. ~~-~~When presented with input sentences and a question, an agent can respond in one of five possible ways:

1. True.
2. False.
3. Input sentences are contradictory, making an answer impossible.
4. Input sentences lack adequate information to answer the question.
    a. In this case, the agent will output a text string requesting "HELP.".
5. Respond to a request for help from another agent by outputting its~~one's~~ knowledge base.

For example, a contradiction would be as follows:

Input ~~s~~Sentences	~A A & B C ==> A
Question	What is C ?
Answer	Contradictory

In thise example ~~above~~, the sentence "~~-~~A & B"~~,~~ implies that A is true, whereas~~but~~ ~~'~~"~A"~~,~~ means that A is false. ~~-~~Owing to this ~~Because there is a~~ contradiction ~~above~~, *any* conclusion may ~~can~~ be drawn ~~from a contradiction,~~ and the knowledge base itself is invalid. ~~-~~In such a case, the ~~an~~ agent needs to report a contradictory state of knowledge from which no conclusion~~s~~ can ~~may~~ be drawn.

In another example, we may have insufficient knowledge:

Input ~~s~~Sentences	~A
Question	What is C ?
Answer	~~Unknown~~ Unknown HELP!

Hence, if a propositional variable is true or false based on the sentences of the knowledge base~~'s sentences~~, an agent should be able to answer true or false. ~~-~~However, ~~But~~ in those cases where the agent's knowledge is either contradictory or insufficient, the ~~an~~ agent needs to report this. ~~-~~In particular, when an agent lacks sufficient information, it should ask another agent or agents~~(s)~~ in the system for help.

Commented [A18]: It seemed that the spaces before the question marks (and later the periods) should be removed, as is standard, unless they are intended to denote a specific meaning. This was not done during editing to maintain consistency with the non-editable spreadsheet entries in the "Data Sets for Training and Testing" section. Please check.

Commented [A19]: Please confirm that this is correct, i.e., that this refers to the "One-Hot Encoding" and "Basic Neural Networks" sections but not the subsequent "Long Short-Term Memory Neural Networks" and "Sequence-to-Sequence LSTMs" sections. As per the earlier comment, it may be useful to add section numbers to allow the sections to be referred to more easily.

Commented [A20]: There appears to be a discrepancy here between "*any* conclusion may be drawn" and "from which no conclusion can be drawn". Please check and consider clarifying if necessary.

Commented [A21]: This is another case where the current layout of the manuscript hinders understanding – this response is not explained to the reader until much later. Please check.

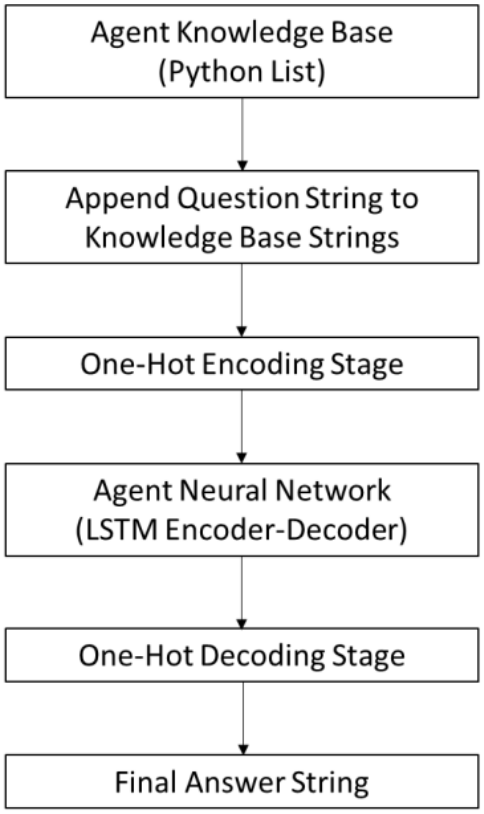


Figure 11: Agent Operation

An agent contains a knowledge base, which is a Python list of strings containing sentences of propositional logic (e.g., ~~i.e.~~ "'A"', '"A ==> B"'). ~~-~~When an agent is presented with a question, the strings from the knowledge base are concatenated with the question string to form the string input, which ~~-~~ ~~The string input~~ is encoded using a one-hot encoding scheme to create an array of inputs. ~~-~~ This input array is then presented to the LSTM neural network (described in the next section). ~~-~~The outputs are one-hot decoded, and the resulting string is returned. ~~We will explain t~~These stages are explained in greater detail below.

1 **Field Code Changed**

Formatted: Do not check spelling or grammar

Commented [A22]: Again, please confirm that this is the intended section.

One-Hot Encoding

Each character in a string, such as a space, a period, a variable name, or an operator or part thereof, is uniquely represented as a column vector with zeros in all positions except one. For example, a straightforward one-hot encoding scheme for the set of letters {a, b, c} could be described as follows:

$$a \rightarrow \begin{bmatrix}1\\0\\0\end{bmatrix} \quad b \rightarrow \begin{bmatrix}0\\1\\0\end{bmatrix} \quad c \rightarrow \begin{bmatrix}0\\0\\1\end{bmatrix}$$

With only three letters to encode, a one-hot vector set with three elements, precisely one of which is one with all the rest being zero, can be used to encode the letters "a", "b", and "c". For the encoding of logical sentences, however, the alphabet has 30 possible characters while the output, which can include extra characters, has 42 possible characters. Hence, each character in an input string is a 30 × 1 column vector with 29 zeros and a single one whose location denotes the character being encoded. Similarly, each output string character is represented by a 42 × 1 vector containing 41 zeros and a single one, the position of which indicates the character of the output alphabet being encoded. Thus, a string of 11 input characters would be represented by a 30 × 11 array, and a string of 17 output characters would be represented by a 42 × 17 output array. Let M denote the maximum allowed length of an input sequence, such that We encode the input as a 30 × M array. If the input is of length L where $L <= M$, then the remaining $M - L$ characters will be blank spaces, represented as one of the 30 possible input characters. If $L > M$, then we truncate at M characters. Hence, all inputs are set to a uniform size of 30 × M, and we denote the input array as $\mathbf{X}$, where the t^{th} column of the input array is denoted as $\mathbf{x}(t)$. The 30 × M array is used to create inputs for the LSTM encoder–decoder neural network and presented to the LSTM encoder stage. At each time step t, where t ranges from zero through (M -1) inclusive, we give $\mathbf{x}(t)$ to the recurrent LSTM sequence-to-sequence network.

The network will generate a set of output vectors $\mathbf{y}(t)$, each of which is a 42 × 1 column. These are concatenated together to create the final output array $\mathbf{Y}$. Because neural network outputs from the softmax operation described in the following two sections are not exactly one or zero (i.e., you may see 0.995 or 0.005 may be obtained instead of 1 or 0), we use the index of the largest element of each output column vector $\mathbf{y}(t)$ to obtain the output character at step t. This process of generating $\mathbf{y}(t)$ given $\mathbf{x}(t)$ lies at the heart of the agent, and an LSTM sequence-to-sequence neural network performs this. The basic aspects of this network are described in the following three sections.

Basic Neural Networks

A basic neural network accepts a vector of inputs $\mathbf{x}$ of inputs and returns a vector of outputs $\mathbf{y}$ of outputs. It normally consists of one or more "dense" layers (Chollet, keras.io, n.d.), each of which can be described by the following equation:

$$\mathbf{x}_{l+1} = f(\mathbf{A}_l \mathbf{x}_l + \mathbf{b}_l)$$

Commented [A23]: For many journals, displayed equations should be numbered sequentially with "(1)", "(2)", "(3)" etc. aligned to the right margin. Please consider applying this style throughout. If so, it may also be desirable to move the equations in Figs. 5–8 outside of the graphics and number them as part of the same sequence.

Commented [A24]: It would be preferable to use either "one" and "zero" consistently or "1" and "0" consistently in such cases, rather than a mixture of both. Please confirm that the result is acceptable.

Commented [A25]: Letters denoting physical quantities/variables (not including vectors) were italicized, as is standard. Please confirm that the result is acceptable.

Formatted: Font: Italic

Formatted: Font: Not Bold

Formatted: Font: Not Bold, Italic

Formatted: Font: Not Bold

Commented [A26]: The bold formatting was removed from "t" for consistency. Please confirm that the result is acceptable or otherwise ensure a consistent style throughout.

Formatted: Font: Italic

Formatted: Font: Not Bold

Formatted: No underline

Formatted: Font: Not Bold

Commented [A27]: These repeated and overlapping references to "the following two sections", "the following three sections", and "later" are becoming somewhat confusing to follow – it is again suggested to consider using section numbers and/or a more clearly defined layout.

Commented [A28]: Please check this citation – it is generally not necessary to include the URL in such citations, but the year should be present. It may also be preferable to cite a peer-reviewed review article or similar here, as opposed to a blog post.

Field Code Changed

1. $\text{relu}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$
2. $\tanh(x)$
3. $\text{sigmoid}(x) = \frac{1}{1+\exp(-x)}$
4. $\sigma_i(x) = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_j)}$

The first three ~~E~~equations ~~(1) through (3)~~ are~~would be~~ applied on an elementwise basis to obtain ~~get~~ the output vector. The fourth ~~E~~ equation ~~(4)~~ defines the softmax function. If **x** is a *K*-dimensional vector then the softmax function is also a *K*-dimensional vector with the i^{th} element defined as in the fourth equation ~~(4)~~. The softmax vector elements are strictly positive and always sum to exactly one~~1~~. In some cases, the identity function $f(\mathbf{x})=\mathbf{x}$ can also be used, particularly for the final dense layer of a multilayer network. If a network has *N* layers, then the final output will be $\mathbf{y} = \mathbf{x}_N$, which is the output of the final layer. ~~Sometimes a~~A different function *f* is sometimes used for the final layer compared with ~~uses a different function *f* than~~ the previous layers.

Neural networks based on one or more dense layers have numerous ~~many~~ applications, and we use a dense layer as part of our larger neural network in a manner ~~way~~ that will ~~to~~ be described later. However, such neural networks lack any memory~~;~~. ~~T~~their current output is strictly dependent~~s strictly~~ on the current input. In the following ~~next~~ two sections, we will discuss a type of recurrent neural network~~, a network~~ whose output depends on both current and previous inputs, i.e., an ~~called a long short-term memory (~~LSTM~~)~~ network.

LSTM~~ong Short Term Memory (LSTM)~~ Neural Networks

LSTM neural networks are among the most successful recurrent neural network architectures for processing series data. ~~Here W~~We provide ~~give~~ a brief description of these networks~~m here~~, closely following the work of Christopher Olah (Olah, 2015), from which we borrow our explanatory figures in this section ~~and briefly summarize here~~.

Traditional neural networks are designed to have a fixed input vector **x** and a fixed output vector **h** that ~~which~~ depends entirely on the input **x**, and such networks are defined by a function $\mathbf{h} = f(\mathbf{x})$. These neural networks have no memory, and **h** does not depend on previous inputs, only ~~on~~ the current input **x**.

A recurrent neural network, in ~~by~~ contrast, has an output that depends on not only ~~on~~ the current input but also ~~on~~ all previous inputs. Let $(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t)$ denote a time series of inputs, and ~~let~~ $(\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_t)$ denote the corresponding time series of outputs. The resulting ~~In a~~ recurrent neural network is depicted in Figure 2~~, we have:~~

Commented [A29]: As per the earlier comment, if displayed equations are to be numbered they should be numbered next to the right margin and all of them should be numbered sequentially. Please confirm this and similar edits or otherwise clarify.

Commented [A30]: As per the earlier comment, "LSTM" should be defined at first use and it has already been used multiple times.

Commented [A31]: Please confirm that the edited sentence has retained the intended meaning. These figures appear to have been taken from a blog post. It is suggested to ensure that any necessary copyright permissions have been obtained. It would also be standard to mention the original source in each figure caption, the wording of which will depend on the source and target journal, e.g., "Reproduced with permission from Christopher Olah's blog (Olah, 2015)."

Commented [A32]: Please confirm that the edited text has retained the intended meaning.

Formatted: Font: Not Bold

Formatted: Font: Not Italic

Formatted: Font: Not Italic

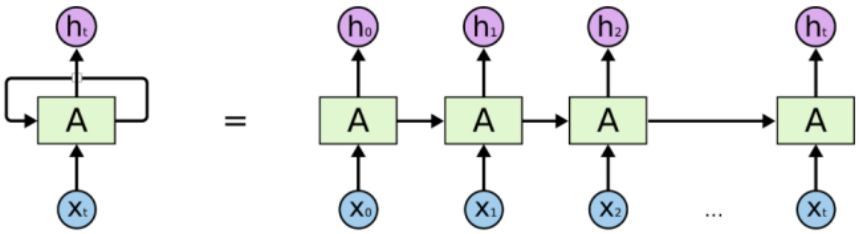


Figure 22: A Rrecurrent nNeural Nnetwork

Here, we observe that $\mathbf{h}_t$ is a function not only of $\mathbf{x}_t$ but of all previous inputs as well. Hence, recurrent neural networks possess have memory, unlike traditional neural networks.

One particularly successful type of recurrent neural network is the LSTM network. The basic architecture of which is shown in Figure 3the LSTM is shown here.

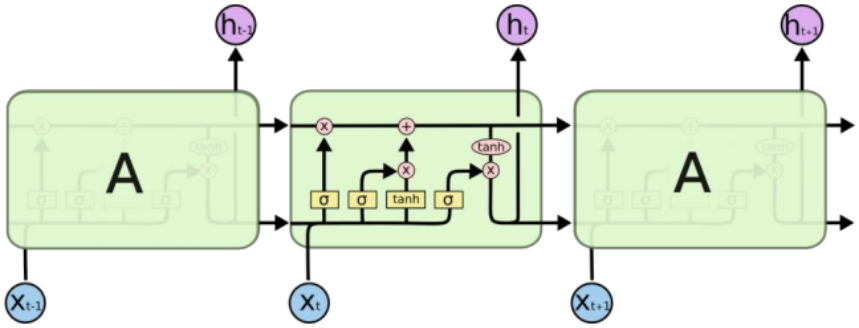


Figure 33: An LSTM Network illustration

We will now explain the internal operation of the LSTM cell in the center middle of Figure 3 Figure 3: LSTM Illustration.

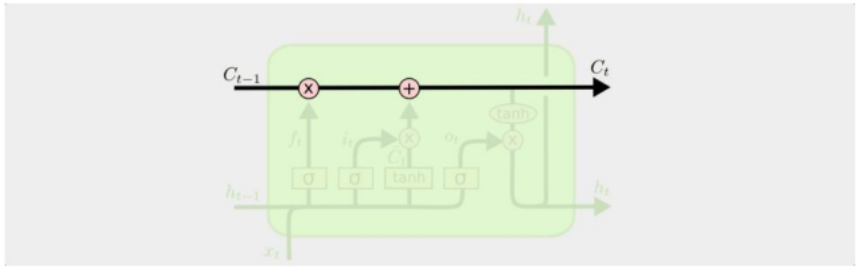


Figure 44: Propagation of Cell State

Commented [A33]: In this and subsequent figures, please ensure that the formatting of the variables is made consistent between the text and graphics, e.g., the bold formatting of "**x**" and "**h**" and the italic formatting of the subscript "*t*".

1 **Formatted:** Do not check spelling or grammar

Field Code Changed

Field Code Changed

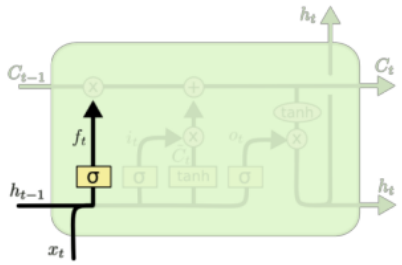
1 **Formatted:** Do not check spelling or grammar

Field Code Changed

6 **Formatted:** Do not check spelling or grammar

Field Code Changed

Here, the state must pass through two key stages: a forgetting stage followed by an updating stage. The forgetting stage multiplies elements of the previous cell state $\mathbf{C}_{t-1}$ by numbers between zero0 (completely forget that vector element) and one1 (remember that vector element perfectly). The subsequent updating stage adds new information to the cell state. The computation of the forgetting stage is shown in Figure 5.~~Figure 5: Forgetting Stage Computation~~



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure ~~5~~5: Forgetting Stage Computation

Once this stage is passed, we must update the cell state. The computation of ~~for~~ the updating stage~~e~~ is presented in Figure 6.~~shown in Figure 6: Update Computation.~~

Figure ~~6~~6: Updating Stage~~e~~ Computation

Combining the above, we see that the new cell state is computed as depicted in Figure 7~~f.ollows:~~

1

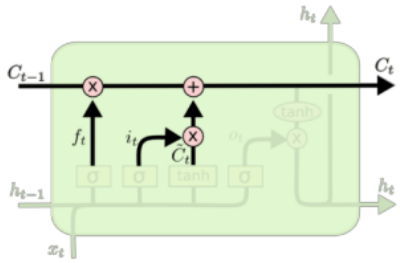
Formatted: Do not check spelling or grammar

Field Code Changed

Field Code Changed

Field Code Changed

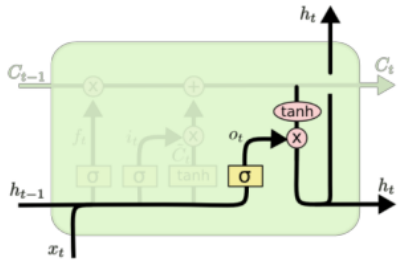
Formatted: Do not check spelling or grammar



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 77: Computing New Cell State

Once the new cell state has been calculated~~computed~~, it is necessary to compute the new output $\mathbf{h}_t$, as shown in Figure 8.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 88: Computing the Output of the LSTM Cell

Sequence-to-Sequence LSTMs

Sequence-to-~~s~~Sequence LSTMs have been successfully applied to numerous ~~used for many~~ complex tasks, including language translation (Bengio, 2014). A description of ~~s~~Sequence-to-sequence LSTMs can be found in (Le, 2014), (Chollet, 2017), and (Bengio, 2014), and our implementation code is a modified version of that~~code~~ from (Chollet, 2017). The description of sequence-to-sequence LSTMs given here summarizes (Chollet, 2017).

1 **Field Code Changed**

Formatted: Do not check spelling or grammar

Formatted: Do not check spelling or grammar

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Commented [A35]: Following on from the earlier comments regarding the citations, it is suggested to change this to "A description of sequence-to-sequence LSTMs can be found in previous works (Le, 2014; Chollet, 2017; and Bengio, 2014), and our implementation code is a modified version of that developed by Chollet (2017). The description of sequence-to-sequence LSTMs given here summarizes that of Chollet (2017)." or similar (note that the exact citation style will depend on the target publication). If possible, it may be preferable to replace these blog posts and arXiv papers with peer-reviewed manuscripts.

Field Code Changed

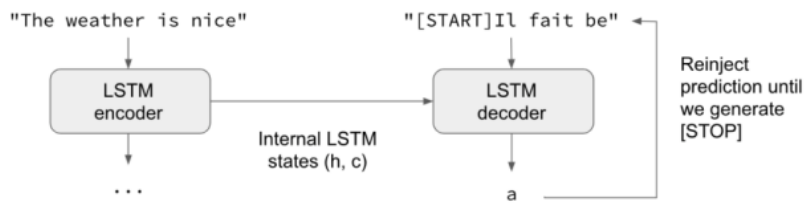


Figure 99: Sequence-to-Sequence LSTM

The process of training such a network is known as teacher forcing. In teacher forcing, which is illustrated in Figure 10, Figure 10: Training a Sequence-to-Sequence LSTM, the input sequence begins with the [START] token and continues one character at a time with the desired French phrase. The desired output sequence is the selected target phrase. **Add links for teacher forcing.**

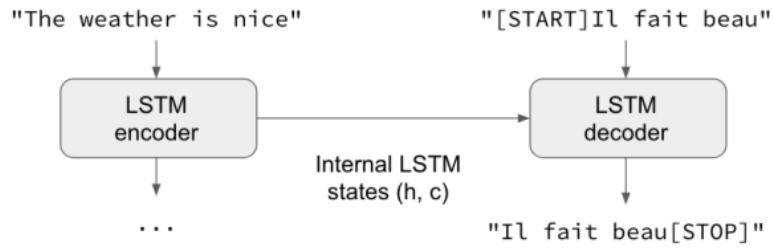


Figure 1010: Training a Sequence-to-Sequence LSTM

The teacher forcing method has been successfully used to train language translation LSTM encoder–decoder networks, and this is the training method used in our work.

Bidirectional LSTMs and Sequence-to-Sequence Models

The sequence-to-sequence LSTM described above is unidirectional, with information flowing from left-to-right in Figure 3, Figure 3: LSTM Illustration. In a bidirectional LSTM, there are two LSTM layers: one layer processes the sequence from left-to-right, while and a second parallel layer processes the sequence from right-to-left in in the opposite direction, as illustrated in Figure 11. We. This is illustrated in Figure 11: Bidirectional RNN below, and we note that bidirectionality can be applied to any type of

1 **Formatted:** Do not check spelling or grammar

Field Code Changed

Field Code Changed

Commented [A41]: It seemed that more information may be needed here – in its present form, this appears to essentially duplicate information already given by (or readily apparent from) the previous paragraph. Similarly, Fig. 10 seems to provide little additional information compared with that already given by Fig. 9 (is the additional figure really justified here?). Please check.

Commented [A42]: Please ensure that this is replaced prior to submission.

1 **Field Code Changed**

Formatted: Do not check spelling or grammar

Commented [A43]: Is a citation needed here? Please check.

2 **Field Code Changed**

Field Code Changed

recurrent neural network RNN, not just LSTMs. We observe an LSTM processes information left to right to decode data, but the second parallel processes the opposite direction.

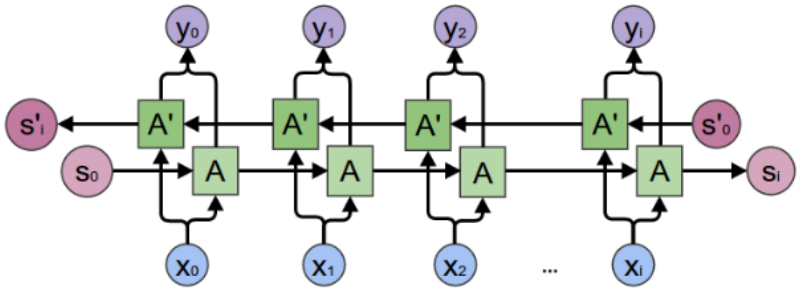


Figure 1111: A Bidirectional Recurrent Neural Network RNN

A bidirectional network allows us to generate output pairs $(\mathbf{x}_i, \mathbf{y}_i)$, where the x's $\mathbf{x}_i$ denotes are the outputs of the left-to-right network and $\mathbf{y}_i$ the y's denotes are the outputs of the reverse (right-to-left) network. The concatenation $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ is a vector whose contents depend on both the previous and the future characters in thea sequence. In cases where the full sequence is available, the resulting concatenated vector encodes information about the complete sequence, not just information from the preceding vious (left) portion of the sequence. Hence, the $\mathbf{z}$'s vectors outputted by that constitute the output of the bidirectional LSTM yield a more comprehensive picture of the entire input sequence than that afforded by the original $\mathbf{x}_i$'s output alone would have alone.

In the sequence-to-sequence model defined previously, we did not n't use the sequence of outputs as an input to the decoder segment of the encoder–decoder architecture. Rather, we simply adopted used the final encoder state $(\mathbf{h}_f, \mathbf{c}_f)$ as the initial state of the decoder network, and we used a start token as the input to go with the initial state. In order t Thus, to implement a sequence-to-sequence model that takes advantage of bidirectionality, we perform do the following steps:

1. Make the encoder stage a bidirectional network. This results in two final states, which are $(\mathbf{h}_f, \mathbf{c}_f)$ for the forward network and $(\mathbf{h}_r, \mathbf{c}_r)$ for the reverse network.
2. Concatenate the states. Here, $\mathbf{h} = (\mathbf{h}_f, \mathbf{h}_r)$ and $\mathbf{c} = (\mathbf{c}_f, \mathbf{c}_r)$.
3. Present this concatenated state $(\mathbf{h}, \mathbf{c})$ to the decoder LSTM as its initial state, in exactly the same manner as as would be done in the previous sequence-to-sequence model.
4. Run the decoder network in thea forward-only direction, just as in the previously presented sequence-to-sequence model.
5. Note that the decoder network state vector dimensionality is now twice that of each of the two encoder networks. Previously, the encoder and decoder networks would have had the same state vector dimensionality.

The neural network adopted used here uses 256-dimensional vectors for $\mathbf{h}_f, \mathbf{c}_f, \mathbf{h}_r, \mathbf{c}_r$. Hence, we have two parallel 256-dimensional LSTMs running in the forward and reverse directions in the encoder stage. This means that the final encoder output state $(\mathbf{h}, \mathbf{c})$ has 512-dimensional $\mathbf{h}$ and $\mathbf{c}$. Because Since this is the initial state for the unidirectional decoder LSTM, the decoder LSTM is based on 512-dimensional vectors.

Commented [A44]: It would also be acceptable to use "RNN" here, but this should then be defined at first use then used consistently throughout.

Commented [A45]: A sentence was removed here because it duplicated information already given immediately above. Please confirm that the result is acceptable.

1

Formatted: Do not check spelling or grammar

Field Code Changed

Commented [A46]: Please confirm the formatting changes made here or otherwise apply a consistent style throughout.

Formatted: Font: Bold, Not Italic

Formatted: Font: Not Italic

Formatted: Font: Bold, Not Italic

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Italic, Subscript

Commented [A47]: Please confirm that the edited text has retained the intended meaning.

Formatted: Font: Bold

Formatted: Font: Italic, Subscript

Commented [A48]: Please confirm that the edited sentence has retained the intended meaning.

Formatted: Font: Bold

Formatted: Font: Italic, Subscript

Commented [A49]: This point seemed somewhat awkwardly placed as a step following the previous four actual steps. Would this perhaps be better presented as part of step 4 or in the text below? Please check.

Commented [A50]: The intended meaning of this text was unclear – does this refer to the concatenated state $(\mathbf{h}, \mathbf{c})$ defined above, or should it be "$\mathbf{h}_f$, $\mathbf{c}_f$, $\mathbf{h}_r$, and $\mathbf{c}_r$" or "each of $(\mathbf{h}_f, \mathbf{c}_f)$ and $(\mathbf{h}_r, \mathbf{c}_r)$"? Please consider clarifying.

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

~~In order t~~To reduce the ~~from a~~ 512-dimensional vector to a one-hot encoded vector, the well-known dense neural network layer described previously, with a final softmax non-linearity function, is used. ~~-~~The index of the maximum element of the output vector tells us which character must ~~is to~~ be decoded.

Data Sets for Training and Testing

The d~~D~~ata sets consist of ~~files are~~ tab-separated files with the filename extension "tsv", with ~~and use~~ tabs used to divide the~~ir~~ data into four columns:

1. Column 0: Original problem and question. ~~-~~This consists of a Boolean sentence, a period, and a question regarding ~~about~~ a Boolean variable. ~~-~~For example, "(~J & B) & F & (B & F) & (B & F) & (B & F) ~~-~~. What is B ~~-~~?" comprises ~~consists of~~ the Boolean sentence "(~J & B) & F & (B & F) & (B & F) & (B & F)" and the question "What is B ~~-~~?". ~~-~~Here, Boolean clauses or variables can be randomly repeated one or more times, and in this example "(B & F)" is repeated three times.
2. Column 1: Simplified problem and question. ~~-~~Here, all repetitions of Boolean clauses, including single-~~one~~ variable clauses, are removed. ~~-~~The sentence is still separated from the question by a period. ~~-~~For example, "(~J & B) & F & (B & F) ~~-~~. What is B ~~-~~?" is the entry corresponding to the sentence above, but with the repetitions removed. ~~-~~Hence, the Boolean sentence that comes before the question has the exact same meaning, but with the repetitions removed it is a concise version of the original sentence. ~~-~~The question remains unchanged from the first column.
3. Column 2: Simplified Boolean sentence. ~~-~~In this example, it would be "(~J & B) & F & (B & F)", which is the simplified sentence from column 1 but without the question.
4. Column 3: The desired answer.~~-~~ The four possible desired answers are:
    a. "TRUE",
    b. "FALSE",
    c. "Contradictory",
    d. "Unknown HELP!",
    e. In this example, the desired answer is "TRUE" because the variable B is TRUE given the sentence "(~J & B) & F & (B & F)".

The above example ~~above~~ would appear in a spreadsheet as follows~~like this~~:

(~J & B) & F & (B & F) & (B & F) & (B & F) . What is B ?	(~J & B) & F & (B & F) . What is B ?	(~J & B) & F & (B & F)	TRUE

All of the training data exist in this four-column tab-separated format.~~-~~ These data need to be one-hot encoded ~~in order~~ to create inputs and targets for the neural network. ~~-~~For each line in the ~~TSV~~ training set, two string pairs are created:.

1. Pair 1: This consists of the original string in column 0 and the target string in column 3.
    a. In the example above, the pair would be ("(~J & B) & F & (B & F) & (B & F) & (B & F) . What is B ?", "TRUE").
2. Pair 2: This consists of the Boolean sentence in column 0, with the question replaced by "HELP", followed by the simplified sentence of column 2.
    a. In the example above, the pair would be ("("(~J & B) & F & (B & F) & (B & F) & (B & F) . HELP","(~J & B) & F & (B & F)").

Commented [A51]: Please confirm that the edited sentence has retained the intended meaning. It is also suggested to add a citation or citations for "described previously".

Commented [A52]: As per the earlier comment, it seemed that the spaces before the periods and question marks here should be removed, as is standard, unless they are intended to denote a specific meaning. This was not done during editing to maintain consistency with the non-editable spreadsheet entry below. Please check.

Commented [A53]: Please confirm that the edited text has retained the intended meaning.

Commented [A54]: As per the previous comment, this seemed awkwardly placed in a list of desired answers (because it is not actually one of the four answers). Would this perhaps be better presented as part of the text below or as an extension of list item 4 on a new line but without "e."? Please check.

Commented [A55]: Please confirm this edit or otherwise clarify – if "TSV" is to be used as an abbreviation, it should be defined.

Commented [A56]: There is a problem with the quotation marks and bracketing here – the two double quotation marks around the first part of the pair appear redundant, an opening quotation mark is missing from the second part of the pair, and one of the two opening brackets at the start is not followed by a closing bracket. Please check.

It may be simpler here to present these on separate lines to avoid some of the need for multiple brackets and quotation marks, and/or use the standard sequence of quotation marks in U.S. English (i.e., double quotation marks for quotations and single quotation marks for quotations within quotations).

The purposes of behind the two training pairs areis as follows:

1. Purpose of PPair 1: Teach the neural network how to answer a question about a Boolean variable when given a Boolean sentence.
    a. If the sentence is contradictory, then no answer is possible, and the network must indicate "Contradictory".
    b. If there is insufficient information, the network should respond with "Unknown HELP!".
    c. If the sentence implies that the variable is true, then the network should respond with "TRUE".
    d. If the sentence implies that the variable is false, then the network should respond with "FALSE".
2. Purpose of Pair 2: Teach the neural network to consolidate a knowledge base, which may contain repetitions, and to dump a concise version of the knowledge base upon receiving when given a requestion for help.
    a. Here, the question is replaced bywith the word "HELP".
    b. Upon seeing the keyword "HELP" instead of a question, the network should create a concise version of the knowledge base without any repetition and dump that concise knowledge base out.
    c. The network must dump out the concise version faithfully without error and without repetition.

Hence, the goal of training is to enable thea neural network to perform two basic functions:

1. Perform Boolean reasoning and answer a question regarding about a Boolean variable given a possibly repetitive, or contradictory, or incomplete knowledge base.
2. Perform simplification and return a concise version of a knowledge base upon given a requestion for help.

The two functions described above are central to agent the operation as depicted in Figure 1of agents described in Figure 1: Agent Operation.. The agent's internal Python list stores Boolean sentences and is the knowledge base of the agent. If an agent is asked a question about the value of a Boolean variable, then the following operations should take place:

1. The sentences of the knowledge base are concatenated using the logical AND operator, which is the ampersand symbol "&".
2. The resulting large Boolean sentence is concatenated with the question following using a period in order to create the input string.
3. The input string is one-hot encoded and fed to the neural network.
4. The neural network output is one-hot decoded to create an output string, which should be one of the four possible responses to a question about a Boolean variable.

Hence, the aforementioned "pair 1" as described above, is used to teach the neural network how to perform the Boolean reasoning operation.

1. The sentences of the knowledge base are concatenated using the logical AND operator (the "&" symbol) as above.

Commented [A57]: It seemed that this somewhat duplicates the previous point and that the two could be combined by changing the latter to "...the network should create a concise and faithful version of the knowledge base without any repetition or error" or similar. Please consider whether this would be appropriate.

Field Code Changed

Commented [A58]: Please confirm that the edited text has retained the intended meaning.

Commented [A59]: Please confirm that the edited text has retained the intended meaning.

2. However, But the large Boolean sentence is now concatenated with the word "HELP", which is separated by a period from the large sentence from step 1.
3. The input string is one-hot encoded and sent to the network.
4. The network output is one-hot decoded and should, ideally, contain a concise and correct Boolean sentence describing the agent's knowledge base, without any repetitions.

Hence, the aforementioned "pair 2", as described above, is used to teach the neural network how to respond to for a request for help by providing a correct and concise dump of the agent's knowledge base.

The Accuracy Metric

Agent performance is measured by simple string comparison. The output of the agent's neural network is one-hot decoded to create a string, which and the string is compared with to the target string. InFor the case of "pair 1", case where the agent must answer a question about a Boolean variable, the output string possibilities are "TRUE", "FALSE", "Contradictory", and "Unknown HELP!". An exact string match is required for the agent's response to be considered correct. For example, if the correctright answer is "FALSE", then the agent's response is scored as correct only if its output string exactly matches "FALSE". Responses such as "F", "false", "False", and or "fALse" would all be scored as incorrect. Similarly In like manner, in for the case of "pair 2," case where the neural network has to perform a knowledge base dump in response to a request for help request, the output must exactly match the target string, which. The target string contains has all of the Boolean clauses of the input string, in exactly the same order, but without any repetition. If the correct answer is ""(~J & B) & F & (B & F)", then answers such as "(~J & B) & (B & F) & F", while clearly logically equivalent, will still be marked as wrong. Hence, in order to achieve a score of 98%, for example, an agent's outputs must be an exact string match to the target 98% of the time, and. Eeven logically equivalent outputs that are not an don't provide this exact match will be scored as wrong. This is a different metric to than the character-by-character accuracy metric, in which that a string with a single character error is marked as wrong, even when all most of the characters but one are correct.

The Data Sets

Each tab-separated file contains 25,000 lines, each of which is used to create both a logic problem, (or "pair 1"), and a problem of restating knowledge concisely (, or "pair 2)". Hence, each file will yield a total of 50,000 training problems split evenly between Boolean reasoning and concise knowledge recitation. The first 100 files, with filenames "logic_data_extended_00.tsv" through "logic_data_extended_99.tsv", are the training files. The remaining files, which are "logic_data_extended_100.tsv" through "logic_data_extended_233.tsv", are for accuracy testing.

Commented [A60]: Please confirm that the edited sentence has retained the intended meaning. In addition, it was not very clear why this alternative metric is being mentioned here – is some more information and/or a literature citation needed? Please consider clarifying.

Commented [A61]: Could this perhaps be combined with the "Data Sets for Training and Testing" section?

Commented [A62]: It was not very clear what is being discussed here – where are these files, how were they generated, and what do they contain? It seems that not enough information has been given to the reader to understand what is being discussed. Please clarify.

Commented [A63]: Some introductory text is needed here to introduce the following numbered list. Please clarify. It may actually be preferable to change some of these lists to regular running text, including the one immediately below and the third one in this section – although these may be acceptable/clearer in some cases, there did seem to be something of an overreliance on them in this editor's opinion.

1. Future neural networks may train on nearly all of these files. The neural network as of repository commit fe2b92d6c7c25b2752650b0358a3df885b794558 from December 29, ~~December~~ 2021, was trained on the first 100 files as stated above. It remains to be seen whether the ~~if~~ accuracy can be improved by training on a larger number of files.
2. The author intends to generate a total of 500 data sets and ~~to~~ use the majority of these to train a new version of the network to determine whether ~~we can achieve~~ improved accuracy can be attained.
3. The author has obtained accuracy exceeding 98% with the existing network for ~~using~~ data set logic_data_extended_200.tsv, a data set not used in training.

It is now important to point out some of the limitations of the data sets in the present work. Although the ~~P~~problems are randomly generated, ~~but~~ a large percentage of the problems, unfortunately, tend to allow a network to deduce a value as true or false simply based on a single Boolean term. This is undoubtedly ~~certainly~~ a weakness of the present randomized clause generator, and it will result in some shortcomings ~~weaknesses~~ when the network is presented with highly complex Boolean sentences. The goal of this work is to focus on the concept of self-awareness, not to create a general-purpose Boolean reasoning system. Even so, the biases with respect to ~~in~~ data set generation are a deficiency that must be addressed in future work. ~~However, t~~The present agents are able to demonstrate the basic~~s~~ criteria of self-awareness that are the goal of this work, but readers are cautioned that they may make errors in Boolean reasoning, particularly for moderately to highly complex sentences~~, the agents may make errors in Boolean reasoning~~. This will require~~s~~ improvements to the data set generation procedures. Data set generation is a random process that can be ~~described~~ summarized as follows~~:~~

1. Boolean variables are randomly selected to populate a sentence.
    a. Some of these are randomly negated to e~~i~~nsure that we have both positive and negative atomic Boolean terms.
2. Binary logic operators, such as logical AND, OR, and implication, are randomly selected ~~in order~~ to create binary clauses.
3. Randomly generated terms and clauses are joined using the logical AND (“&”) operator to create sentences.
4. A variable is randomly chosen for the question, and each question is always about the value of a single variable, which may be negated.
5. The resulting sentences are processed using automatic reasoning code, taken from (Norvig, aima-python, n.d.), ~~in order~~ to determine whether the variable is ~~t~~True or ~~F~~false, or whether the input knowledge ~~re~~ is insufficient ~~information~~ or ~~whether the input knowledge is~~ contradictory.
6. The clauses of the sentence~~s~~ may ~~can~~ be repeated one or more times ~~in order~~ to generate the “long” version~~s~~ of the sentences ~~that may contain repeated clauses~~.
7. The “long” sentence, with repetition, is concatenated with the question to create the column 0 entry.
8. The “short” sentence, without repetition, is concatenated with the question to create the column 1 entry.
9. The “short” sentence, without repetition, becomes the column 2 entry.
10. The answer to the question (e.g., ~~i.e.~~ “TRUE”, “FALSE”~~, etc.~~) becomes the column 3 entry.

The choice to include ~~have~~ repetition in the data set design requires some explanation~~:~~

Commented [A64]: Again, it was not very clear what is being discussed here. Which repository is this? Please clarify.

Commented [A65]: The intended meaning of this statement was unclear without a prior description of how the data sets were generated. Please check and consider clarifying if necessary.

Field Code Changed

Commented [A66]: Please see the earlier comment regarding this citation.

1. In earlier attempts, Boolean reasoning performance without the repeated clauses proved to be poor. For example, if an agent saw a clause repeated twice, it could make errors in reasoning.
    a. In multi-agent scenarios, if a given sentence was known to two agents, the repetition of the sentence would sometimes result in reasoning errors. This was the original motivation for our repetitive clause training.
2. Forcing agents to learn how to create a concise sentence with each clause present repeated only once is believed to teach their internal neural network to treat clauses as conceptual units, resulting in improvements to reasoning performance.
    a. This is a hypothesis, however, that requires further testing, and the author does not claim sufficient evidence to assertclaim that this type of training actually does teach a neural network to treat terms and clauses as conceptual units.

All of the TSV data set files are available at (Mukai, S3 Source Code and Data Sets, n.d.) in order to facilitate peer review and to make both the strengths and the weaknesses of the training and test data easily accessible and understood.

Key Software Used and Neural Network Specifications

The source code is publicly available at (Mukai, LSTM Source Code, n.d.) and is based on previously developed source code for LSTM neural networks from (Chollet, keras.io, n.d.) and Boolean logic from (Norvig, aima-python, n.d.) with the LSTM neural network code taken from the former and with the Boolean logic code taken from the later. The training and testing process works as follows:

1. The program train_seq2seq_help.py is used to:
    a. Create the neural network itself.
    b. Perform a training epoch on logic_data_extended_00.tsv.
    c. Save the resulting neural network.
2. The neural network created is a bi-directional LSTM with
3. The program retrain_seq2seq_help.py is used to:
    a. Load the neural network created in step 1 above.
    b. Run for a specified number of epochs, (presently 512).
    c. On each epoch:
        i. Randomly select a training set from logic_data_extended_00.tsv through logic_data_extended_99.tsv.
        ii. Perform a training epoch using the randomly selected data set.
4. The program run_seq2seq_help.py is used to:
    a. Load the neural network.
    b. Load the file logic_data_extended_200.tsv for use as a test set (this was not used in the training above).
    c. Compute the neural network accuracy over this data set logic_data_extended_200.tsv.

4. The program run_seq2seq_demo.py is not a standalone program.

Commented [A67]: Please see the previous comments regarding such citations – the title in particular should not be present inside the citation, a year should be included, and this in any case seems to be missing from the reference list.

Field Code Changed

Commented [A68]: It was not very clear what this means – assuming that this manuscript is intended for submission to a journal as a research article, references to the peer review process would not normally be made in the article itself (although they may be made in a cover letter). Would it perhaps be better to refer to the readers here instead? Please check.

Commented [A69]: Please see the previous comments regarding such citations – the title in particular should not be present inside the citation, a year should be included, and this in any case seems to be missing from the reference list.

Field Code Changed

Field Code Changed

Field Code Changed

Commented [A70]: Please confirm that the edited text has retained the intended meaning.

Commented [A71]: This point is incomplete. Please clarify.

Web Demo

The goal of the Google Colab web demo is to provide a simple demonstration of agent self-awareness, which, again, is defined as an agent's ability to be aware of its own knowledge state, (in this case being aware of not knowing the right answer), and acting according, (in this case by requesting aid from another agent).

In the web demo, :

~A

C ==> B

Agent 2 begins with just one fact:

B ==> A

Agent 1 is presented with the following question:

What is C ?

However, agent 1 does not possess sufficient information to ascertain the value of C. At this stage, agent 1 asks for help, causing agent 2 to dump its knowledge base. Agent 1 is now armed with all three facts:

~A

C ==> B

B ==> A

Thus, agent 1 can reason that because A is false and B ==> A, then B must be false. Because B is false, then C ==> B implies that C is also false. The demo ends with agent 1 indicating that C is false.

Agent 1 demonstrates a very simple form of self-awareness in terms of being aware of its own lack of knowledge. When agent 1 realizes that it lacks the knowledge to answer the question regarding Boolean variable C, it asks for help. Likewise, agent 2 also demonstrates a simple form of self-awareness in that it knows what its knowledge base contains and will dump those contents in response to a request for help. It should be noted that this self-awareness is a property of the agent, which is a composite of a neural network plus a Python list knowledge base and appropriate code that performs one-hot encoding and decoding and can generate or receive a request for help. Hence, the neural network, while certainly the most important part of the agent, is not the entire agent. Thus, self-awareness is a property of the complete agent, not a property of the neural network as a standalone entity.

Summary and Conclusions

This work presents agents that exhibit a simple form of self-awareness defined by the following:

Commented [A74]: The italics for emphasis here seemed unnecessary – they appeared to be more distracting than helpful in this editor's opinion. Please confirm that the result is acceptable.

Formatted: Font: Not Italic

Commented [A75]: This is another example of a numbered list that, in this editor's opinion, may be better presented as regular text. Please consider whether this would be suitable.

1. Knowledge of one's own knowledge base.
    a. This is ~~E~~exemplified by the ability to provide a concise version of its ~~the~~ knowledge base upon a ~~on~~ request for help from another agent.
2. Knowledge of one's own knowledge state.
    a. The ability to know whether one's knowledge is contradictory.
    b. The ability to know whether one's knowledge is insufficient.
    c. The ability to answer a question when one has sufficient and non-contradictory knowledge.

~~A possible variation of this work may use a neural network to guide a formal reasoning engine.~~ Because ~~Since~~ Boolean reasoning already entails substantial computational complexity, and considering that ~~since~~ well-known neural networks have been able to excel in handling problems with exponential complexity, most notably the game of Go (DeepMind, n.d.), ~~a~~, a promising variation of this work ~~direction~~ may involve using neural networks to guide a formal reasoning engine. This will help to prevent outright errors in reasoning, a weakness of the present work, while simultaneously helping to overcome the exponential complexity of reasoning, which can be a significant problem with epistemic modal logic.

Hence, this work represents ~~is~~ a starting point. The form of self-awareness described ~~presented~~ here is extremely basic, but ~~and~~ the ~~work~~ presented work should serve as ~~is~~ a proof of concept to demonstrate the potential of neural-network based systems to exhibit a basic form of self-awareness.

Commented [A77]: This citation appears to be missing from the references section. Please check.

Field Code Changed

Commented [A78]: Two sentences were combined here to avoid repetition. Please confirm that the result is acceptable.

References

Bengio, K. C. (2014, September 3). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* Retrieved from arXiv: https://arxiv.org/abs/1406.1078

Chollet, F. (2017, September 29). *A ten-minute introduction to sequence-to-sequence learning in Keras.* Retrieved from The Keras Blog: https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html

Du, B., Guo, Q., Zhao, Y., Zhi, T., Chen, Y., & Xu, Z. (2020). Self-Aware Neural Network Systems: A Survey and New Perspective. *Proceedings of the IEEE*, 1047--1067.

Evans, R., Saxton, D., Amos, D., Pushmeet, K., & Grefenstette, E. (2018). *Can Neural Networks Understand Logical Entailment?* Retrieved from arXiv: http://arxiv.org/abs/1802.08535

Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics.* Retrieved from arXiv: https://arxiv.org/pdf/1912.01412.pdf

Le, I. S. (2014, June 3). *Sequence to Sequence Learning with Neural Networks.* Retrieved from arXiv: https://arxiv.org/abs/1406.1078

Mukai, R. (2022, February 11). *Dual Agent Demo*. Retrieved from Dual Agent Demo: https://colab.research.google.com/drive/1Tr2AQKTGtG3VnNgHhRFLt2BXiXqSb3No?usp=sharing

Commented [A79]: Because the references are still present as field codes from the reference management software, they were not directly edited. A separate version without the field codes has been pasted after this section and some minor changes have been made for internal consistency.

However, there was a limit as to what could be done at present without a target publication – many of the sources cited are somewhat non-standard (i.e., blog posts or similar, rather than peer-reviewed journal articles), the formatting of which will heavily depend on the target publication. As a general rule, these will likely need to be formatted as online sources with as many details as possible and a retrieval date (this may also apply to the arXiv papers). As noted in the text, it may also be preferable to replace some of these citations with references to peer-reviewed literature where possible, and it seemed likely that the references list will expand after the "Introduction" section has been written.

It was also indicated during manuscript upload that the focus for this round of editing should be the technical content rather than formatting, and it was assumed that this will need to be dealt with during a subsequent round of editing when a target publication has been selected.

Field Code Changed

Norvig, S. J. (2021). *Artificial Intelligence: A Modern Approach, 4th Edition.* Hoboken, NJ: Pearson Education, Inc.

Norvig, S. J. (n.d.). *aima-python.* Retrieved from aima-python: https://github.com/aimacode/aima-python

Olah, C. (2015, August 27). *Understanding LSTM Networks*. Retrieved from colah's blog: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Bengio, K. C. (2014~~, September 3).~~ *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* Retrieved from arXiv: https://arxiv.org/abs/1406.1078

Chollet, F. (2017~~, September 29~~). *A ten-minute introduction to sequence-to-sequence learning in Keras.* Retrieved from The Keras Blog: https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html

Du, B., Guo, Q., Zhao, Y., Zhi, T., Chen, Y., & Xu, Z. (2020). Self-Aware Neural Network Systems: A Survey and New Perspective. *Proceedings of the IEEE*, 108, 1047~~–~~–1067.

Evans, R., Saxton, D., Amos, D., Pushmeet, K., & Grefenstette, E. (2018). *Can Neural Networks Understand Logical Entailment?* Retrieved from arXiv: http://arxiv.org/abs/1802.08535

Lample, G., & Charton, F. (2019). *Deep Learning for Symbolic Mathematics.* Retrieved from arXiv: https://arxiv.org/pdf/1912.01412.pdf

Le, I. S. (2014~~, June 3~~). *Sequence to Sequence Learning with Neural Networks.* Retrieved from arXiv: https://arxiv.org/abs/1406.1078

Mukai, R. (2022~~, February 11~~). *Dual Agent Demo*. Retrieved from Dual Agent Demo: https://colab.research.google.com/drive/1Tr2AQKTGtG3VnNgHhRFLt2BXiXqSb3No?usp=sharing

Norvig, S. J. (2021). *Artificial Intelligence: A Modern Approach, 4th Edition.* Hoboken, NJ: Pearson Education, Inc.

Norvig, S. J. (n.d.). *aima-python.* Retrieved from aima-python: https://github.com/aimacode/aima-python

Olah, C. (2015~~, August 27~~). *Understanding LSTM Networks.* Retrieved from ~~colah's~~Colah's blog: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Commented [A80]: It is generally not necessary to include actual dates in such citations. Because these were not present consistently for all arXiv citations, they were removed. Please confirm that the result is acceptable or otherwise ensure consistency throughout.

Commented [A81]: Please note the added volume number.

Commented [A82]: It is suggested to cite the arXiv papers consistently, e.g., see the use of the PDF link here and the "abs" link elsewhere.

Self-Aware LSTM-Based Agents

ORIGINALITY REPORT

2%
SIMILARITY INDEX

PRIMARY SOURCES

1	li.mit.edu Internet	109 words — 1%
2	Anne Cocos, Alexander G Fiks, Aaron J Masino. "Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts", Journal of the American Medical Informatics Association, 2017 Crossref	17 words — < 1%
3	stteresaschoolpawtucket.com Internet	14 words — < 1%
4	www.24x7editing.com Internet	12 words — < 1%
5	digitalcommons.dartmouth.edu Internet	11 words — < 1%
6	documents.mx Internet	11 words — < 1%
7	dspace.cuni.cz Internet	9 words — < 1%

EXCLUDE QUOTES OFF EXCLUDE SOURCES OFF