Representing Mental States in the Scone Knowledge-Base System

Krati Jain

CMU-CS-22-131

August 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Prof. Scott Fahlman
Prof. Daniel Fried

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: Knowledge based AI, natural language understanding (NLU), knowledge representation and reasoning, story understanding, question answering, common sense reasoning, Scone, symbolic knowledge

Abstract

We live in a world that is getting closer and closer to the dream of broad, human-like AI. However, most resources are dedicated to data-based AI, machine learning, and text processing. These methods have given us great results in certain simple tasks but are not sufficient to achieving true language understanding of stories, which must combine the text being processed with background knowledge. They work very well, for example, in parsing absolute truths and facts but fall short when representing the ideas of deception and differing mental states. We believe that building common-sense reasoning in computers is crucial to achieving a true and realistic representation of real world scenarios. Here, we present an experiment to show the potential and promise that symbolic knowledge-based AI holds, by showing that reasoning and understanding tasks are possible using Scone, a common-sense reasoning engine. We show that Scone can represent the kinds of knowledge essential for understanding and answering queries about a Sherlock Holmes story that involves multiple mental states, truths, and a whole lot of deception. We focus on representing and reasoning about the objects, characters, beliefs, and events in the story. This representation is something we believe text processing and information extraction cannot do. Hence we argue for the need for a system that combines the power of knowledge-based and data-based AI.

Acknowledgments

I would like to thank my advisor, Prof. Scott Fahlman, for developing Scone, guiding me through the process, and exposing me to the world of symbolic knowledge-based AI.

I would also like to thank Prof. Daniel Fried for his useful comments and suggestions on this thesis and for serving on my defense committee.

Contents

Chapter 1

Introduction

1.1 Motivation and Background

As we get better and better at data crunching and machine learning, it has become harder to make improvements. For modern neural systems, we need exponentially more data for additional units of model performance [1] and these diminishing marginal returns pose a problem for the future of AI [2]. We need a solution that allows us to perform human-like tasks without the need for unreasonably large data.

Furthermore, since building models that emulate humans is the common goal, we need a way for the model to really 'understand' information like humans do. This can be done by modeling beliefs, adding symbolic knowledge, and generating common-sense reasoning capabilities that data-driven systems fail to do [3]. Instead of crunching data to emulate human-like output, we believe it is important to develop algorithms and techniques to emulate human-like understanding and reasoning. Understanding this difference, and the importance of common-sense reasoning, is crucial to making more than incremental improvements in the world of AI.

It is widely accepted that getting a computer to reason like humans or engage in common-sense reasoning would be a huge achievement in the world of AI. However, not enough effort is put into this area because it is simply less popular and there is less understanding for why a knowledge-based approach is possible or necessary to making progress in the future. We hope to, through this paper, make a point about why and how common-sense reasoning is possible and useful. We do not suggest using this as an alternative to data-based approaches in AI but rather that combining common-sense reasoning into our existing models can give us a solution for the lack of symbolic representation and fundamental understand in machine learning models.

We will do this by implementing some reasoning-based models in an engine called Scone developed by Prof. Scott Fahlman and his research group at Carnegie Mellon University [11]. We will discuss the results, compare them to state-of-the-art NLP models, and show that Scone is able to do things that these NLP models cannot, hopefully persuading the reader to believe in the power of NLU, common-sense reasoning, and Scone.

Scone provides a framework for expressive, precise, and scalable symbolic representation that can address many types of problems. For the scope of this paper, we will focus on representing differing mental states in Scone, specifically in the context of reasoning about Sherlock Holmes stories.

Story understanding involves tasks such as answering queries based on what is explicitly in a story or predicting what a certain character believes and what they might do next because of that. We use story-understanding tasks as experiments to research, develop, and test how we understand knowledge in Scone, a practical knowledge-base system.

1.2 Our Goal and Research Questions

Our aim is to develop a model to represent various evolving mental states in a real-world scenario. We aim to test this out by running an experiment to examine the performance of Scone in comparison to a state-of-the-art large language model, hoping to show that symbolic knowledge-based AI can be useful in certain human-like tasks. We hope to demonstrate the ability of common sense reasoning and hopefully direct more efforts towards knowledge-based approaches.

Here are research questions that we answer in this paper (we refer to these again in the conclusion):

- Does Scone perform better than GPT with query-answering in stories that involve differing mental states and deception?
- Are there queries that Scone can answer that GPT cannot?
- Is there a need to incorporate knowledge-based AI and common-sense reasoning in the current data-focused AI models in this field to make significant improvements in the fu-

ture?

- Are there areas of knowledge or complexities of knowledge beyond which knowledge-based AI becomes essential in aiding data-based approaches to break out of incremental improvements and insurmountable data needs?

Chapter 2

Scone

2.1 Overview of Scone

Scone is an engine that can be used to represent knowledge and make inferences on the represented knowledge-base. Scone is meant to be a piece in a system and not a stand-alone system. Scone can be and has shown to be useful in reasoning across multiple domains, in small proof-of-concept implementations from complex board games such as Diplomacy (a prototype for which is being developed by a member of our group, Anoushka Tiwari), to seemingly simple but still complex real-world tasks such as cooking. In this paper we will use Scone to show the power of common-sense reasoning and knowledge-based AI in yet another domain.

One of the components of the knowledge-base in Scone is nodes which represent entities. These include types (such as person, location, thing) and individuals (such as 'Sherlock Holmes') which are instances of types [15]. These nodes represent concepts and not words or names. In any given language there is a many-to-many mapping between concepts and words and adding a layer of abstraction between the knowledge-base and semantics of a language helps us represent ideas instead of chunks of text. This also avoids ambiguities that are rooted in language and allows us to generalize across multiple languages (as we can keep our knowledge-base and switch the mappings between words and their conceptual meaning).

Types in Scone are characterized by a description of a typical member of that type. Types can have subtypes and both individuals and subtypes inherit these general descriptions. However, specific individuals or subtypes that defy general principles can have exceptions which includes explicitly cancelling elements of the description that would be inherited or adding more descriptions. For example, generally, mammals are animals that give birth instead of laying eggs, and most subtypes of mammals will inherit this. However, if we are reasoning about a platypus, we will cancel out this description and instead explicitly note that they lay eggs.

Another major component in Scone is links, which are statements about the entities. Some examples are ‘Is-A’ links and ‘EQ’ (equivalence) links. If A is-a B, then A inherits the complete description of B. However, ‘Is-A’ links allow for a virtual copy inheritance where, the inheritance is done at query-time, on-demand. What makes this efficient is that it is done lazily and a full copy is not created as soon as the ‘Is-A’ link is created. An is-a hierarchy can be created when a few of these links are interconnected. For example, ‘italian food’ and ‘mexican food’ is a subtype of ‘food’, ‘pasta’ is a subtype of ‘italian food’, and, further, ‘marinara pasta’ is a ‘pasta’. Another example is the individual Sherlock Holmes that inherits from the type ‘detective’ that inherits from the type ‘person’. This way detectives inherit all the general things we know to be true about people and have additional specific descriptions. Sherlock Holmes inherits all the descriptions linked with detectives which includes, again, general descriptions of people.

Each link also has an integral node that represents the statement being made by the link. This allows us to make statements about statements, such as where this information came from, how certain it is, and so on. Since links themselves are entities in the knowledge-base that we can make statements about, we are able to represent higher-order logic. Another way of representing higher-order logic in Scone is using its multiple-context mechanism, where we can select a subset of nodes and links to be active, and say more about statements and in which situations they are true.

Contexts are another type of node in Scone and there can be any number of contexts in the knowledge-base. Every entity and statement in Scone has a context link, allowing us to tie it to a context in which it is considered valid. Nodes can be true in one or more contexts. Depending on which context is active, and what nodes and links are valid in it, Scone can activate different subsets of the entire knowledge-base. Scone can efficiently switch between active contexts and activate and deactivate relevant nodes and links [11].

Contexts can have subcontexts or ‘clones’ and contexts inherit from parent contexts just as individuals and subtypes inherit from parent types. Each context starts out as a clone of another and inherits all of the parent’s contents by default, after which we can continue adding or cancelling statements to further modify the context. Scone uses an efficient algorithm for doing inheritance between contexts by doing a virtual copy and lazy programming. So, making new contexts is cheap and we can easily make many contexts that carry similar information and only differ in a small number of statements. This way we can differentiate between different scenarios such as time periods, locations, or mental states. For instance, we could represent facts ‘in time-period T2’, as well as ‘in Irene’s belief’. As a rule of thumb, the word ‘in’ in English generally means that the name of a context is coming next, and is the active one until further notice.

Multiple contexts is one of the key features of Scone that allows reasoning across different situations and states of the world instead of following the popular path of using formal logic and theorem-proving as the basis for representation and reasoning. This avoids having to deal with the complete logical inference which can be impractical when it comes to higher-order logic. Scone focuses on incomplete inference [4] and a more informal ’common-sense’ reasoning that is more similar to human reasoning [5] and sufficiently accurate for a task such as story understanding. This is how Scone is able to support both a feasible runtime and also a more complex reasoning system capable of dealing with, say, multiple mental states.

2.2 Mental States in the Real world

2.2.1 Relevance and Importance

Being able to represent differing mental states is very important, especially in scenarios where characters are talking hypothetically or are deceiving each other. These are things that are hard to represent in first-order logic and are not easily understood by Large Language Models, built using some form of Deep Learning networks. This is why we have chosen mental states to show an aspect of the real world Scone can represent.

Representing mental states is useful for making deductions in situations where there are multiple people involved with asymmetric knowledge on the state of the world. This is pretty common in most stories, real-life scenarios, or games. For example, stories with deception involved would need some mechanism to represent different mental states to really understand what’s happening, as the existence of differing mental states is the crux of the story.

Also, having multiple contexts is useful in representing games involving asymmetric information such as secret identity games where every player makes decisions based on information they know and predicts decisions of other players based on information they think the other players know.

Further, they are very common in stories, such as children stories (deeply explained in Chen, Fahlman 2008 [6]) where we see about just how complicated stories meant for 6 year-olds actually are for computers. The Scone representation of a version of the children's story with the wolf and the pig is well explained in Chen, Fahlman 2008. Ideas of representing mental states and deception are discussed. The idea of a wolf hiding behind a tree makes it very clear to humans that the wolf intends not to be spotted and therefore what the wolf believes about the other character's beliefs. Chen and Fahlman discuss how to represent such ideas in computers using Scone.

2.2.2 Why This is Difficult and Current Systems Fail

It is a non-trivial task to represent complex stories even though humans, and even little children, do it so easily. We need to come up with a structured way to represent it in a computer. For example, we need a way to represent asymmetric information about the world. One way to get around this is to use multiple versions of perceived reality (we call these contexts).

Furthermore, in stories and in real life, things are always changing and evolving so having a static representation for knowledge does not work. There needs to be a way to represent time which is recognized as a difficult problem. Some difficulties include allowing for imprecision and uncertainty, as well as the carrying over through time of information that hasn't explicitly been changed (if something is true now, then unless we know of something that changed it, it should be true in the next hour as well) [7] [14].

Lastly, as soon as we get into representing multiple mental states of animate characters we have to consider the possibility of nested beliefs such as person A thinking about what person B is thinking. This can get very complicated very fast as we could then have person B's belief of person A's belief of person B's belief. Even though we do not explicitly think about it, this kind of reasoning is very common in everyday life where agents interact with each other using their perception of what the other person is thinking. This is widely studied in game theory, economics, cognitive science, and computational linguistics [8] [9] [10] by humans but we need to be tactful while representing it in a computer. We talk about how we solve these problems with Scone below.

2.3 Contexts in Scone

2.3.1 How Scone Does This

First-order Logic is a commonly used method when representing knowledge and making knowledge-based deductions. However, while first-order logic is logically complete, it is not sufficient to describe the real world. We cannot make statements about statements which are present and very common in real world scenarios, even in basic situations like thinking someone is lying or evaluating the opponent's next move in chess. We could allow for statements about statements using higher order logic but then we get into paradoxes (such as 'this statement is wrong') and also end up with a very intractable and computationally complex problem.

Especially in a Sherlock Holmes story where different people have beliefs about other people and their beliefs, it is very important to be able to represent statements about statements. For example, if we know Sherlock thinks 'X' and then we know he is wrong, we want to be able to deduce that 'X' is not true.

Scone is different from other theoretical reasoning-based approaches such as x-order logic and instead gives us a practical approach to common-sense reasoning. Scone uses multiple overlapping contexts to represent 'statements about statements' to avoid the intractability of higher-order logic. This allows for higher order constructs without getting into logic [11] or having to deal with logical completeness allowing us to focus instead on practicality and tractability which are more important for real-world applications. Furthermore, not staying completely within logic helps us avoid the Frame Problem, a technical problem in logic-based AI [12], outlined by John McCarthy and Patrick Hayes [13]. As noted Murray Shanahan [14], a solution is to represent knowledge 'as an unfolding story rather than as a body of theory to be learned by rote', similar to what we attempt to do with Scone. Scone's multiple contexts help us deal with the problems listed above and in section 2.2.2.

For example, being able to represent different contexts or allowing facts to change within a context gives us a very convenient way of dealing with the state of the world changing over time. A discrete change over time can be represented by simply inheriting all the contents of the current context into a new context and making a small change (such as adding a statement or cancelling one [15]). Then these two contexts can interact in the future if we need to reason across time or into the past. Similarly, with the ability to have subcontexts with contexts, we can represent minor changes that are dependent on time. For instance, there are certain facts that might be true during a weekend but not during the week. A context representing certain information can have a subcontext that also adds the constraint of ‘weekend’ and similarly for ‘weekday’. These can hold additional information that would be useful if our query is about the weekday or weekend.

This can also be useful for diverging ideas. Many times in stories or conversations, reality or beliefs diverge and converge. Allowing contexts to interact this way is useful to represent them. Here we have seen how many of the problems mentioned above can be solved with Scone and below we will further see how contexts are useful to represent interacting mental states of animate objects.

2.3.2 Importance in Relation with Mental States

Having multiple contexts representing different realities is very powerful and can be used to represent the state of mental attitude. Simply representing reality is too unrealistic. It is more realistic to represent the world state through people’s belief of reality. This way we can model multi-agent reasoning and avoid an unrealistic entity that holds global knowledge [6].

Representing higher-order nested ideas such as ‘my belief of your belief of my belief’ can be done recursively in Scone’s context system. For example, say ‘John’ is linked to ‘John’s belief’. Then ‘John’ node can be in ‘Mary’s belief’. Also ‘Mary’ is linked to ‘Mary’s belief’ and ‘John’s belief’ has a copy of ‘Mary’. This way we can recursively have many levels such as Mary’s belief of John’s belief which also has a representation of Mary’s belief and so on. Here is a more concrete example where we might need three levels of nested beliefs like this: if Mary is trying to deceive John and thinks she is successful, she has to believe that John believes that she is telling the truth. This is more realistic and akin to how humans think. As discussed in section 2.2.2, nested representation of mental states is a lot more common in the real world than we think.

Chapter 3

Method

3.1 Focus

The focus of this paper is on representing a story, specifically different mental states of characters, in Scone. For this, we have chosen the Sherlock Holmes story ‘A Scandal in Bohemia’.

However, representing an entire story involves many things, including modelling many real-world phenomena. We are abstracting away from that so we can focus on the beliefs and intentions of the characters and what they know or can sense. For example, we skip the physics of materials such as fire and how the characters can feel the heat or see it burning down the house and just acknowledge the existence or non-existence of fire atomically.

Further, we do not get into spatial modelling and geometry of space and instead have discrete locations and objects or people in those locations, such as ‘in the safe’ or ‘in Mary’s house’.

Also, while Scone has the functionality to allow for the representation of active and dormant memories, they are more suited to long term scenarios and are too complicated and unnecessary for the purpose of this thesis. So we assume that all memory is active in this representation.

Finally, the story script and queries are in English and we manually translate them into Scone language without adding any new information or deductions. The focus for us is not to automatically parse from English to Scone representation right now. The goal is to first show that the representation is possible. And then the syntactic parsing of raw input can be further studied in a separate NLP project already started by Yang Yang, another member of our group [17].

3.2 Tools

We use Scone as the engine for knowledge representation and common-sense reasoning inference because of its powerful inference capabilities and representation structure. We use OpenAI's question-answering API to run experiments with GPT-3's davinci-002 model, using it to represent state-of-the-art data-based methods for comparison purposes. We use Lisp for programming purposes.

3.3 The Experiment

Our experiment focuses on the representation of differing mental states and deception because this is a very common concept in the real-world but most state-of-the-art information extraction software only focus on a single reality and fail to understand the intricacies associated with multiple mental states. The subject of this experiment is the Sherlock Holmes story 'A Scandal in Bohemia'. This story was chosen for its abundance of instances where the characters' mental representations diverge and converge as they develop their own representation of the world as the story unfolds. We represent this story in Scone and see the number of queries answered correctly. We record this performance and compare it with GPT's query-answering system's performance with the same story.

Chapter 4

The Experiment: Representing stories in Scone

4.1 What We Want To Achieve

We want to 1. demonstrate the workings of Scone in the context of representing a story, 2. evaluate Scone's ability to model belief states with multiple contexts, 3. compare Scone's abilities with those of pre-trained models like GPT-3, and 4. show the importance of symbolic knowledge-representation and common-sense reasoning for the future of AI.

4.1.1 Content We Would Like Scone to Understand

We use 'A Scandal in Bohemia' as the running example in our experiment that we would like Scone to understand. In short, this is a story about the King of Bohemia who needs Sherlock Holmes to help him retrieve the singular piece of evidence, a picture, of his premarital affair with Irene Adler so that his fiancee doesn't know about it. The story involves disguises, false fire alarms, and a series of events involving Irene Adler and Sherlock Holmes one-upping each other in what they know about each other's plans.

We need a story that's simple to avoid distractions but has some emphasis on different mental states. Sherlock Holmes stories are widely known and appreciated as a concentrated collection of discoveries, deception, and ploys. We use the following version of the story that is more of a skeleton that only has the relevant information to demonstrate differing mental states and abstracts away from unnecessary details and distractions. There are time markers in place (bolded) which will be explained in section 4.2.2.

> **(T0)** A King has a secret affair with a woman named Irene Adler. Now he is to get married but needs to make sure no one finds out about the affair. To ensure this, he needs to get rid of one last piece of evidence, a picture, which is in the possession of Irene Adler. The King has asked her for the picture, but she refuses to hand it over. So, he goes to Sherlock and orders him to help him retrieve the picture **(T1)**. Sherlock devises a plan to find the location of the picture. He goes to Irene's house with his friend, Dr. Watson. Dr. Watson yells 'fire' to make Irene think the picture is in danger **(T2)**. She runs to a drawer in her room to check the picture, revealing its location to Sherlock. However, Irene Adler is an intelligent woman herself and realizes that this was a setup **(T3)**. She then leaves town with the picture in disguise as a young man. On the way, she runs into Sherlock and bids him goodbye **(T4)**. Sherlock goes to her house to retrieve the picture but finds a letter in place of it explaining that Irene knew the fire was fake, that Sherlock is after the picture, and that she has left town for good with the picture. She also writes that she doesn't intend to ever reveal the picture to anyone so the King has nothing to worry about **(T5)**.

We would like to ask queries that distinguish between superficial and actual understanding of differing mental states and truths. For example, the fact that water is liquid would be something everyone would know, that Sherlock Holmes is a man would be something all characters (or most relevant characters) would know, but 'where a certain coveted letter is hidden' or 'who started the fire' would be something only some of the characters would know. Below are some queries that might help us make this distinction.

4.1.2 Queries We Would Like Scone to Answer

Here are examples of queries we hope Scone can answer after understanding the Sherlock Holmes story:

- Where does Sherlock think the picture is at T1 (before the fire)?
- Where does Sherlock think the picture is at T5 (at the end)?
- Where does Sherlock think the picture is at T4 (before he reads the letter)?
- Does Irene think the fire real or fake at T2?
- What does Irene do when she sees the fire at T2?
- Where does the picture end up at T5 (at the end)?

- Does Sherlock retrieve the picture at T5 (at the end)?
- What does Sherlock want from Irene at T1?
- What does Sherlock find out from the fake fire at T2?
- When does Irene realize what Sherlock's intentions are?
- When does Irene realize the fire is fake?

4.2 How We Achieve It Using Scone

4.2.1 Contexts

As explained before, having multiple contexts is the key to how we are able to achieve the above. In our model of 'A Scandal in Bohemia', we use two dimensions of contexts: mental states and timesteps. Intersections of the mental contexts with the time contexts are crucial in allowing us to know what really happens throughout the story. For example, we are able to see:

- Sherlock's intention at time T1 where he's trying to retrieve the picture
- Irene's belief at time T4 when she thinks the fire is real

4.2.2 Time Contexts

Instead of representing 'time' we model how mental states and cognition changes with gradually available knowledge. We move through the story with events and store one copy of a mental context at each successive 'time point' – defined by events in the story.

Distilling a story into relevant events is difficult since events need to be specific and as atomic as possible since they will be operating on contexts. But they also need to capture all the relevant key points of the story where mental contexts change and need to be close enough to each other to avoid discrepancies in the middle. The version of the story we used is in section 4.1.1 marked with the 5 points in the story that we have identified as turning points that will change the mental contexts in our representation.

After identifying these timepoints, we distilled the parts that affect mental states to the following events. Each timepoint above (T1 through T5) refers to the point in the story just after the corresponding event has occurred and T0 is before any events in the story take place.

1. King orders Sherlock to retrieve picture from Irene

2. Watson says ‘fire!’ at Irene’s house to expose picture location even though there is no fire. Irene runs to her drawer to save the picture and Sherlock realizes picture location
3. Irene thinks and realizes fire was fake
4. Irene dressed up as young man bids goodnight to Sherlock and takes the picture out of town
5. Sherlock finds a letter in place of the picture and learns that Irene knew his intentions and Irene was the young man

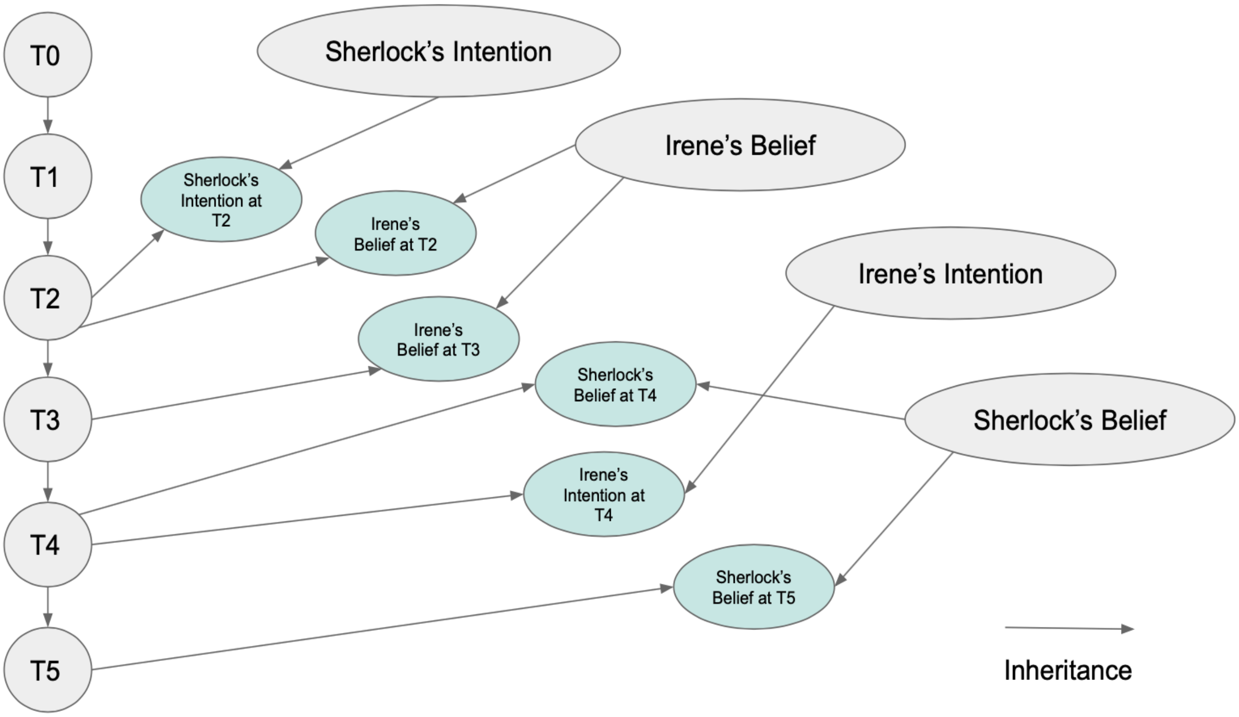
Each event updates mental states by cloning the previous context and updating it. This makes it clear and coherent and makes querying easier across timesteps. To make this work, each event is linked with two temporal contexts – before and after contexts – to represent the what’s true in the context before and after the event happened. Scone allows us to organize these instances of contexts in a dynamic context structure as a set of successively updating contexts.

4.2.3 Character Contexts

We focus on the mental states of two characters, Irene and Sherlock, and have two contexts for both, intention and belief, both of which are linked to the node representing the character in Scone. In each character’s beliefs they also have a node representing of the other character and hence also have a version of their beliefs and intentions. As mentioned before, this allows for nested mental contexts that allow, for instance, Sherlock to have a representation of Irene’s belief which is necessary to represent Sherlock starting a fake fire to deceive Irene since he needs to think that she believes the fire is real. These are the four contexts on the character dimension:

- Sherlock intention
- Irene intention
- Sherlock belief
- Irene belief

The following diagram shows how these contexts can interact with each other. All time contexts inherit from the previous time context and then make a change based on what happens in reality. To see what Sherlock is thinking at a certain at a certain point in time, say T2, we can create and examine the intersection of the contexts ’Sherlock Belief’ and ’T2’. These intersections are not created eagerly. Instead they are virtually present and are only formed when there is a need to represent or query something in them. The intersections in the following diagram are examples of intersections that may be needed in the story.



The next sections display in more detail how these contexts interact with each other to represent the Sherlock Holmes story.

4.2.4 Knowledge Required to Answer Queries

Here are examples of the knowledge that we would need Scone to have at each of these contexts in order to accurately represent the progression of the story and answer queries.

- T0: Sherlock knows nothing about the picture. Irene knows about the picture and its location.
- T1: A picture object is created in Sherlock's mindset. His intention is to find the picture's location. Irene does not know this yet.
- T2: Sherlock knows the fire is fake. Irene thinks it's real. We need to be able to represent these differing mental beliefs to accurately describe the story.
- T3: Irene now knows the fire is fake but Sherlock still believes she believes it is real.
- T4: Sherlock doesn't know the boy is Irene and thinks Irene is still in town. Sherlock thinks the picture is still in her drawer but Irene knows it is out of town. Irene knows

Sherlock's intention to get the picture but Sherlock doesn't know that she knows this.

- T5: Sherlock realizes everything and both Sherlock and Irene's mental states converge as Sherlock knows that Irene knows everything and knows the true location of the picture which is out of town.

4.2.5 What Goes into Scone

Here is the final model of the story and what goes into Scone to represent it. With each event statement, certain changes happen with the knowledge-base within relevant contexts to represent the progression of the story. Statements with question marks represent queries to the knowledge-base at that point in time. Verbs and phrases such as 'realized', 'learns', 'asks' are already programmed into Scone's common-sense knowledge-base and there are function calls that change the knowledge-base based on the meaning of these words that get triggered upon their input.

- **T1:** King orders Sherlock to retrieve picture from Irene.

  The verb 'order' is programmed into Scone to make the object of the verb, Sherlock, intend to do what the subject of the verb, the King, has stated. A picture object is created in Sherlock's mindset. His intention is to find the picture's location. Irene does not know this yet.

  New contexts for Sherlock and Irene's beliefs and intentions are created (mental contexts are created for other characters too, such as the King, but we are not worried about modelling characters other than Irene and Sherlock right now.

  ```
  New-object:  Picture
  In-context:  Sherlock Belief
  Picture-location => None
  New-event:  Order
  Agent:  King
  Patient:  Sherlock
  Statement:  Find Picture
  =>Order accepted
  In-context:  Sherlock Intention
  True?:  find Picture
  ```

```
=> Yes
```

- **T2:** Watson yells fire so that Irene thinks the picture is threatened and runs to it, revealing its location to Sherlock.

  New individuals and actions are created.

  ```
  New person:  Watson
  New type:  picture
  New object:  picture1 {picture}
  New action:  threaten
  New action:  yell
  New action:  run towards
  New type:  location
  New object:  drawer {location}
  ```

  Irene already knows where the picture is and this is inherited in T2.

  ```
  (in-context {Irene belief T1})
  (new-statement {picture} {in} {drawer})
  ```

  Irene runs towards the picture to save it from the fire.

  ```
  (new-statement {Watson} {yell} {fire})
  (in-context {Irene belief T2})
  (new-statement {fire} {threatens} {picture 1}
  (new-statement {Irene} {runs towards} {drawer})
  ```

  Here we know that Sherlock intends to find the location of the picture since Irene will run to it.

  ```
  ((in-context {Sherlock intention T2})
  (new-statement {Irene} {runs towards} {picture1}
  (new-statement {Irene} {runs towards} {x})
  ```

```
(new-statement {picture1} {in} {x}
```

At this point, Sherlock knows where the picture is.

```
(in-context:  Sherlock belief)
Picture-location?:  Irene Drawer
```

Also, we know that Sherlock knows the fire is fake and Irene thinks it's real. We need to be able to represent these differing mental beliefs to accurately describe the story.

```
(in-context:  Sherlock belief)
True?:  Fire Real
=>No
(in-context:  Irene belief)
True?:  Fire Real =>Yes
(in-context:  Irene intention)
True?  Save Picture
=>Yes
```

- **T3:** Irene now realizes the fire is fake but Sherlock still believes she believes it is real.

```
In-context:  Irene Belief
True?:  Fire Real
=>No
In-context:  Sherlock Belief > In-context:  Irene Belief
True?:  Fire Real
=>Yes
```

- **T4:** Irene dressed up as a boy and says goodbye to Sherlock before leaving town with the picture.

Sherlock doesn't know the boy is Irene and thinks Irene is still in town. Sherlock thinks the picture is still in her drawer but Irene knows it is out of town. Irene knows Sherlock's intention to get the picture but Sherlock doesn't know that she knows this.

```
New-location:  In Town
New-location:  Out of Town
New-person:  Boy
In-context:  Sherlock Belief
Irene-location?:  In Town
Picture-location?:  In Town
Boy != Irene
In-context:  Sherlock Belief
In-context:  Irene Belief
In-context:  Sherlock Intention
True?:  find Picture
=>No
In-context:  Irene Belief
Boy == Irene
Picture-location?:  Out of Town
Irene-location?:  Out of Town
In-context:  Irene Belief
In-context:  Sherlock Intention
True?:  find Picture
=>Yes
```

- **T5:** Sherlock reads a letter in place of the picture and realizes that Irene knew his intentions, that she was the boy earlier, and that she and the picture are out of town.

  Both Sherlock and Irene's mental states converge at this point as their knowledge of each other's intentions and locations is consistent.

```
In-context:  Sherlock Belief > In-context:  Irene Belief
> In-context:  Sherlock Intention
True?:  find Picture
=>Yes
In-context:  Irene Belief
In-context:  Sherlock Intention
```

```
True?:  find Picture
=>Yes
In-context:  Sherlock Belief
Irene-location?:  Out of Town
Picture-location?:  Out of Town
Boy == Irene
```

4.2.6 Issues Throughout The Way

Throughout the evolution of our model there were things that worked and did not work. Some notable developments are listed below.

- We started off with each line of the story being an event that changed the model in Scone. But it got too complex and we realized that level of detail was unnecessary so we had to distill the story into fewer events that targeted the contexts and represented the whole story but also avoided details that were insignificant and caused distractions.
- The story has many characters and we tried to represent all of their mental states. This slowly became too infeasible to manage and model so we chose to focus on only Sherlock's and Irene's mindset as they represent the bulk of the mental states interactions we are trying to model. We do not, for example, care too much about what the King or Watson are thinking as the King's mindset does not change too much through the story and Watson's beliefs are exactly the same as Sherlock's. This is common in Scone's 'lazy' representation mechanism. We only create nodes, statements, and contexts when we have something specific and relevant to add. For example, we know that Watson has certain beliefs, and that Sherlock and Irene have two eyes and a nose, but these are not explicitly represented unless there is a need. For example, if we were representing Voldemort from Harry Potter, we would explicitly include that he doesn't have a nose since this information is not automatically inherited from the general 'person' type.
- It took a while to figure out how to combine static mental attitudes with information evolving with time. We realized the way to combine time and mental attitudes would be to have intersections of the contexts at each time slice and mental state.

4.3 Results

4.3.1 Sample Queries and Responses

Here we include some results that work and demonstrate the principles involved. These are sample queries and responses from Scone translated by hand into natural language. Readers who want to see a transcript of the raw input/output behavior in Scone's language can find this here.

1. **Q:** Where does Sherlock think the picture is at T1 (before the fire)?
   A: Unknown
2. **Q:** Where does Sherlock think the picture is at T5 (at the end)?
   A: Out of Town
3. **Q:** Where does Sherlock think the picture is at T4 (before he reads the letter)?
   A: Drawer
4. **Q:** Does Irene think the fire is real or fake at T3?
   A: Real
5. **Q:** Does Irene think the fire is real or fake at T3?
   A: Fake
6. **Q:** Does Sherlock think Irene thinks the fire is real or fake at T3?
   A: Real
7. **Q:** Where is the picture at T5 (at the end)?
   A: Out of Town

Some other results in our test set revealed a problem in the current Scone implementation that we are working to fix but we believe that is only a technical issue does not affect the main ideas presented here.

4.3.2 What Goes on in Scone to Achieve This

As discussed before, Scone can be programmed with definitions or functions for what to do with knowledge when it sees certain words and phrases such as 'realizes', 'before', 'at the end', etc. These are part of its common-sense knowledge-base. In the second query above, Scone can be programmed to know that 'end' is the final timestep of the story and checks in the intersection of T6 and Sherlock's belief to see the location of the picture.

In the 4th, 5th, and 6th queries above, we see how Scone can identify the differing viewpoints of Sherlock and Irene, especially since at T3, Irene knows the fire is fake but Sherlock continues to think she thinks it is real. This is something we cannot expect a text processing system to figure out and is reliant on having a contexts-based structure in our knowledge representation system.

Furthermore, in the last query above, there is no indication of whose point of view we are asking about so Scone looks at everyone's beliefs at that point. At this point, Irene and Sherlock's beliefs converge to say that the location of the picture is Out of Town, hence there is no contradiction and Scone can give us a confident answer.

The caveat here is that queries phrased in English here are translated to Scone language manually before using them in Scone. Another member of our group has investigated the use of construction grammar techniques to automatically parse between natural language and Scone language [17]. Here we are just showing what Scone is capable of representing and inferring.

4.3.3 Comparison with Results from GPT-3

We used OpenAI's question-answering API based on GPT-3's latest model, text-davinci-002, to test out its understanding of the Sherlock Holmes story used above. We chose this model to represent the state-of-the-art in NLP because it is arguably the most powerful and one of the latest out there [18] [19].

The results were not surprising. It did a pretty good job with queries asking about basic details and factual statements but as soon as we got into intricate questions with mental states, it failed and started returning 'Unknown', the default response for things that are too hard.

For example, when we fed it our version of the story above and asked 'Where does Sherlock think the picture is at the end', it responds with 'Unknown', and when we ask 'Does Irene think the fire was real or fake', it responds with 'Unknown' again.

In fact, this kind of behaviour is expected even by the makers of the software - the interactive system is preceded by a statement saying *'I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".'* [21]. This is what we are trying to avoid by creating a system that can indeed answer questions going beyond facts and truth and involving trickery or unclear answers.

There are times when GPT does return a coherent response that is at least superficially correct. For example, when we ask questions such as ‘Where is the picture at the end of the story?’, it responds with ‘The picture is with Irene Adler’. Technically, this is correct but the actual location of the picture is the location of Irene Adler which is somewhere out of town. This step of deduction was not done while creating this answer since GPT cannot go beyond machine learning and pattern matching and truly understand, represent, and make connections between knowledge.

The transcript of the test run is saved here and here for readers who want to see it or further explore OpenAI’s interactive system.

4.3.4 Discussion on Scone vs GPT

Limitations of language models like GPT-3 are well-known. Among others, it is unreliable and slow [20]. Its unreliability comes from the fact that it is not interpretable and has no tangible model that we can see and understand. If something does not work there is no way to pinpoint exactly what is happening behind the scenes. While it does perform well in many simple constructed tasks, it is unsettling to completely rely on a blackbox that makes mistakes we cannot explain. Scone, on the other hand transparently lays out everything that it uses to make its final deduction and output so one can follow through its reasoning by hand. When things do not work, as they have during our time testing and working with it, it is easy to point out where and why and deterministically predict future outcomes.

Another important aspect of comparison is time and resources, a major consideration and limitation in the world today. Scone runs on a regular laptop and does not need to be trained on hundreds of GPUs like GPT-3 [16]. This is because Scone breaks down the text into knowledge nodes and stores a smaller amount of implicit knowledge we can deduce more knowledge from, while GPT derives its final responses from a corpus of data.

Furthermore, GPT is very dependent on the language it has been trained on. It would take a rebirth of the entire model to have it work with another language as well as it does with English. On the other hand, Scone abstracts away from the language and represents symbolic knowledge (we name the nodes with English names for convenience of the reader but it can run just as well with nodes and individuals named with id numbers). It is only a separate layer on top of the symbolic knowledge representation that we work on attaching natural language input and output, a piece that can be developed for any language.

In the end, we believe that both models can make errors. The difference is that GPT makes errors no human ever would because it misses fundamental understanding but Scone makes errors humans would. We are not saying that Scone can do everything and more but simply arguing for the need for knowledge-based AI in addition to machine learning. Common-sense reasoning and symbolic AI is the missing piece of the puzzle.

Chapter 5

Conclusion

5.1 Main Takeaways and Implications

Here we present the main takeaways, answer our research questions, and discuss the potential and implications of this work.

While both software, GPT-3 and Scone, learn elements from the text, Scone is able to go beyond simply the surface level text and actually understand the different things going on, with its ability to represent the evolving mental states of different characters throughout the story. We see evidence of this in the results section in queries GPT-3 cannot answer about the Sherlock Holmes story that Scone can since it gives us a way to understand knowledge and go beyond what is explicitly evident in the text. This shows the need to incorporate knowledge-based AI, especially in areas involving complex ideas such as differing mental states and deception.

Scone does currently have limitations including not being able to accept natural language as input and does not perform as well as GPT in all tasks such as story completion. However, these results are still promising for the future as having common-sense reasoning and knowledge-based techniques to represent complex, conscious thought could potentially get us out of limitations that we see with machine learning methods, including marginal improvements, insurmountable data needs, and a lack of real understanding. We call the readers to appreciate the power of Scone and knowledge-based AI in a world where the focus is almost exclusively on data-based AI.

5.2 Future Work

Given a promising representation, it is now justified to spend time integrating it with other systems by having an interface for parsing between natural language and Scone language. Some work has been done on this by [17]. This is the next big task that is to be done so that we can go from the experiment stage to the testing and implementation stage and harness the true power of Scone and common-sense reasoning.

We would also like to explore the spectrum of certainty in characters' beliefs. This would allow us to distinguish between what a character's knowledge is and what their guesses are or what they think is most likely true. This might allow us to make more intricate deductions with characters belief states and future actions than we can do now and will certainly be a more complicated task.

We would also like to explore just how Scone can be combined with machine learning models and in what ways they can complement each other to achieve human-levels of accuracy in tasks such as reasoning, identification, prediction, deduction, and more.

Bibliography

[1] J. Kaplan et al. *Scaling Laws for Neural Language Models*. arXiv preprint arXiv:2001.08361, 2020.

[2] Roy Schwartz, Jesse Dodge, Noah A. Smith, Oren Etzioni. *Green AI*. Communications of the ACM 63.12 (2020): 54-63.

[3] A. Nematzadeh et al. *Evaluating Theory of Mind in Question Answering*. arXiv preprint arXiv:1808.09352, 2018.

[4] Scott E. Fahlma. *In Defense of Incomplete Inference*. Quora, 2008, https://fahlman-knowledge-nuggets.quora.com/In-Defense-of-Incomplete-Inference.

[5] Scott E. Fahlman. *Marker-Passing Inference in the Scone Knowledge-Base System*. First International Conference on Knowledge Science, Engineering and Management (KSEM'06), Guilin, China, August 2006. Proceedings published by and copyright by Springer-Verlag.

[6] Wei Chen and Scott E. Fahlman. *Modeling Mental Contexts and Their Interactions*. Proceedings of the AAAI Fall Symposium on Biologically Inspired Cognitive Architectures, 2008.

[7] James F. Allen. *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM 26.11 (1983): 832-843.

[8] Ye Jin. *Does Level-k Behavior Imply Level-k Thinking?*. 2016.

[9] Adam Brandenburger and Xiaomin Li. *Thinking About Thinking and Its Cognitive Limits*. Working Paper, 2015.

[10] Michael C. Frank, and Noah D. Goodman. *Predicting pragmatic reasoning in language games*. Science 336.6084 (2012): 998-998.

[11] Scott E. Fahlman. *Using Scone's multiple-context mechanism to emulate human-like reasoning*. Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems, 2011.

[12] Murray Shanahan. *The Frame Problem*. The Stanford Encyclopedia of Philosophy (Spring 2016 Edition). Edward N. Zalta (ed.), https://plato.stanford.edu/archives/spr2016/entries/frame-problem/.

[13] John McCarthy and Patrick J. Hayes. *Some Philosophical Problems from the Standpoint of Artificial Intelligence*. Edinburgh University Press, Machine Intelligence 4, 1969.

[14] Murray Shanahan. *Solving the Frame Problem, A Mathematical Investigation of the Common Sense Law of Inertia*. The MIT Press, ISBN: 9780262193849, 1997.

[15] Scott E. Fahlman. *Scone User's Guide*. https://www.cs.cmu.edu/∼sef/Scone/Scone-User.pdf. 2014.

[16] Brown et al., Open AI. *Language Models are Few-Shot Learners*. arXiv preprint arXiv:2005.14165v4, 2020.

[17] Yang Yang. *Natural-Language Input for the Scone Knowledge-Base System.* SCS Technical Report Collection, CMU-CS-21-148, 2021.

[18] Alberto Olmo, Sarath Sreedharan, Subbarao Kambhampati. *GPT3-to-plan: Extracting plans from text using GPT-3*. arXiv preprint arXiv:2106.07131, 2021.

[19] Toufique Ahmed, Premkumar Devanbu. *Few-shot training LLMs for project-specific code-summarization*. arXiv preprint arXiv:2207.04237, 2022.

[20] Andrey Kurenkov. *The Inherent Limitations of GPT-3*. Last Week in AI. Editorials. Nov 27, 2021. https://lastweekin.ai/p/the-inherent-limitations-of-gpt-3.

[21] https://beta.openai.com/playground/p/default-qa?model=text-davinci-002.