LLMs for Indexing Stories in a Case-Based Reasoner

Nischal B K, Sathya N C, Ravi Regulagedda
Luddy School of Informatics, Computing and Engineering,
Indiana University, Bloomington IN 47408, USA
nbangal@iu.edu, satnc@iu.edu, raregul@iu.edu

Abstract

The indexing problem deals with the issue of how to best index a case in memory for the most efficient retrieval. While there are many theories of indexing, indices based on the content of the cases provide the most benefits for retrieval. In this paper, we explore a way to use LLMs to directly generate indices for stories. We present a framework for prompting the LLM to generate UIF indices and found that with just a little context and guidance, the generated indices map very well onto the real indices. Our prompting framework is general and can be used to create indices for distinct types of stories. This system can be used as a component in CBR-based story explainers. Finally, we also present two ways for testing the performance of these generated indices quantitatively.

Introduction

The performance of any Case-Based Reasoner depends on its ability to retrieve the best possible case that "reminds" it of the problem at hand. All other factors being equal, for a retriever to pick the best case, it should have access to all the features of the case that are salient to the problem at hand, and the domain being explored. This leads us to the indexing problem.

At its core, the indexing problem refers to the issue of how to best index cases in memory. In an ideal scenario, cases are indexed by their entire content thus enabling the retriever to know exactly what is contained in each case. However, it is highly inefficient since any retriever would have to parse the entire content of each case. This would also occupy too much memory since (depending on the retriever), the case might have to be present twice, once as itself and once as its index.

To resolve this problem, we should use a framework that can hold the most salient parts of the contents of a case. This is called the content theory of indexing, where the main basis of the indices are the features of contents of their respective cases. The UIF is a framework developed at Northwestern University (Roger Schank et. al. 1990) to act as a general content theory-based index to support a range of CBR applications. The main idea driving the development of this indexing framework is to support CBR systems in retrieval of stories.

In this paper, we will focus on generating UIF indices for a story explainer case-based reasoner using LLMs. The task of creating indices for stories is one that requires natural language understanding and reasoning capabilities and is usually done by human experts. The focus of this paper is to explore and present the efficacy of using the knowledge-base present in LLMs and their reasoning to generate indices from stories written in pure English.

The UIF Specification

The Universal Indexing Framework (UIF) is a "frame", a collection of slots. The complete frame acts as the index of the case. The main idea behind the UIF specification is to have an index that is ideal for retrieval. Therefore, the slots that make up this index are features that describe the contents of the case.

The UIF index consists of three main parts: global slots, context grid and content grid. Together, they describe the most salient features of the story that is being indexed while each part focuses on a specific type of feature to be represented. The idea behind story explanation systems is to focus on anomalies and use them as points of focus. This would require the indices themselves to also hold the features of anomalies within the stories they represent. This means the parts of the UIF are primarily representing the different anomalies present in a story. They are explained here in brief.

The global slots consist of two slots that together describe the "setting" and the primary anomaly of the story. Together, they provide a high-level description of the story. The context grid is a table establishing the main actors in the story and their relationships. The content grid then describes the perspective of the different "viewers" of the story as described by the context grid.

This entire perspective of a single viewer in the content grid is called an "intentional chain" (Roger Schank et. al.

1977), and it describes the line of reasoning of the viewer. The slots needed to describe the intentional chain of one viewer are- viewer, view, agent, anticipatory effect, pre-task belief, task, theme, goal, plan, result, complex-result, positive side-effects & negative side-effects.

Together, all these slots along with the actors-relationships and the global slots are required to index a story in the UIF framework. Not all the slots are filled in all the cases since each index ultimately depends on the content of the story and the views of the actors and observers in it. Given the complexity of this indexing framework and relatively "primitive" reasoning capabilities of LLMs, we decided to focus on generating only a subset of the full UIF specification.

Refined UIF Specification

The refined UIF specification we use as the framework to generate indices in was developed by Robin Burke et. al. (1994) for indexing stories in a tutor system that used stories to teach social skills (Robin Burke 1993). This framework uses more generalized slots in the content grid and focuses only on the most salient anomaly in the story.

The reduced content grid consists of the following six slots- theme, goal, plan, positive side-effect & negative side-effect. These reduced slots still form a proper intentional chain and thus can be used to index the story within the same content theory as the full UIF specification.

Experimentation

The language model ChatGPT was taught the structured UIF index via natural language explanation of each of its parts, which was tailored to encapsulate specific story details. To facilitate this training, a content grid outlining the notable Chicago demolition story was provided. This story can be found in the box Story 1. The data was initially formatted with rows and columns defining key attributes, followed by the entire UIF specification for comprehensive training. Our approach aims to refine and augment the model's capacity to understand and respond to similar topics or inquiries in future interactions.

In the subsequent phase of testing, the language model ChatGPT was presented with the Monty Python story to evaluate its comprehension. The model generated an index in response. While ChatGPT demonstrated an understanding of the story's context, it also populated numerous UIF slots unnecessarily. This outcome highlights both the model's contextual grasp and its tendency too overpopulate certain data fields within the structured format.

> Once while watching the demolition of a building in Chicago, I was struck by how ineffectively the work was being done. The wrecking ball hit one of the concrete supports near its center again and again with little result. It was frustrating to watch the lack of progress. This poorly-executed demolition reminded me of the time I saw a bullfight in Spain. The matador kept dealing out blows to the bull with his sword with seemingly little effect. The failure to execute a "clean kill" made the whole affair grotesque.

Story 1: Chicago Demolition Story

> Our lab holds internal research seminars with a slight twist: "Friday Fights" lack the usual rules of decorum, and vitriolic attacks against presented ideas are not merely allowed, they are encouraged. At one Fight, the presenter was unusually tenacious. Despite strong, often devastating, arguments against his ideas he refused to capitulate. His bluster and insistence on fighting on began to seem humorous. This reminded one observer of a scene in a Monty Python movie. Two knights are engaged in a duel. One refuses to admit defeat even after losing both arms and legs even after being decapitated. He insists that the other knight continues the fight or else calls it a draw.

Story 2: Monty Python Story

After informing the language model that not all slots needed to be filled and requesting it to leave spaces for unnecessary slots, ChatGPT showed improvement in its performance. However, some errors still occurred during the process, indicating that further refinement is necessary to enhance the model's accuracy and efficiency in handling structured data inputs. We began to contemplate more effective prompts to provide to the language model. The exact details of these prompts are present in the appendix document attached to the report.

As discussed above, using a restricted index can help ensure that more information is retained accurately in the correct slots within the language model. So, we trained the SPIEL (Structured Prompt for Information Extraction and Learning) Restricted UIF and then tested it on the next story to evaluate its performance. The UIF indices and stories used in our experiments are briefly described below.

Slot names	*View1*	*View2*	*View3*	*Anomaly*
Viewer	Self	Self	Self	
View	perceived	expected	perceived	
Agent	Self	Operator	Operator	
Anticipatory affect	curious			
Pre-task belief				
Task		execute	execute	
Theme				
Goal		demolition	demolition	
Plan				
Result		fast demolish	slow demolish	failed expectation
Positive side-effect				
Negative side-effect				
Resultant affect	bored			
Post-task belief				
Delta affect	-invert			
Delta belief				
Trade-off				

Table 1: Chicago Story UIF

Slot names	**View1**	**View2**	**View3**	**Anomaly**
Viewer	Observer	Observer	Observer	
View	perceived	expected	perceived	
Agent	Presenter	Presenter	Presenter	
Anticipatory affect	intrigued			
Pre-task belief				
Task	participate in seminar	participate in seminar	participate in seminar	
Theme	research seminar	research seminar	research seminar	
Goal	engage in intellectual debate	engage in intellectual debate	engage in intellectual debate	
Plan	present ideas	present ideas	present ideas	
Result	strong defense	strong defense	humorous atmosphere	refusal to admit defeat
Positive side-effect				
Negative side-effect	vitriolic attacks encouraged	vitriolic attacks encouraged	vitriolic attacks encouraged	
Resultant affect	amused	amused	amused	
Post-task belief				
Delta affect	+amused	+amused	+amused	
Delta belief				

Table 2: Monty Python Story UIF prompt no.1

Slot names	**View1**	**View2**	**View3**	**Anomaly**
Viewer	Observer	Observer	Observer	
View	perceived		perceived	
Agent	Presenter			
Anticipatory affect	intrigued			
Task	participate in seminar			
Theme	research seminar			
Goal	engage in intellectual debate			
Plan	present ideas			
Result	strong defense		humorous atmosphere	refusal to admit defeat
Negative side-effect	vitriolic attacks encouraged			
Resultant affect	amused			
Delta affect	+amused		+amused	

Table 3: Monty Python Story UIF prompt no.2

Slot names	**UIF values**
Physical setting	demolition, Chicago, building
Social setting	bull-fight, Spain
Viewer	observer
Perspective	watching
Agent	wrecking ball, matador
Anomaly Type	lack of progress, failure to execute a "clean kill"

Table 4: Restricted UIF

Slot names	**Wanted**	**Actual**
Theme	ineffective work, frustrating lack of progress, grotesque affair	ineffective work, grotesque affair
Goal	demolition	execute "clean kill"
Plan	hit concrete supports, deal blows with sword	hit concrete supports, deal blows with sword

Result	lack of progress	little effect
Side+		
Side-	frustration	

Table 5: Extension of Restricted UIF

Video: I was in the South Bend/Mishawaka area. This was my first or second year. I was calling on a swimming pool contractor. He had quarter page in South Bend. I was proposing a quarter page in another directory. It was sitting at his kitchen table. And the guy was hesitating; he didn't know... So, after a few more attempts, he says to me "OK, I'll go with the other quarter page." He bought it. I pushed the order form in front of him. He signed it. It's done. As I'm putting my stuff together, I made this comment that cost me the quarter page. I said, as I'm packing up, "I'm sure that you're going to get a lot of phone calls from that new ad." He looked at me and he said, "You know, I don't need any more phone calls. I'll tell you what, let's not do that this year, maybe next." I talked myself out of a quarter page. I've never done it since. I walked out. There was nothing I could say. I had it and I lost it. All I had to say was "Thank you very much Joe. See you next year." But I didn't. I had to tell him about the calls, which I'd already done twenty times.1 Nothing bad happened to you because you kept talking to the client after the sale was closed, but sometimes the client changes his mind.

Story 3: Talking myself out of a sale

Results

We designed 5 different iterations of the prompt, each providing varying levels of details regarding the task. As expected, the slots predicted by the LLM (large language models) improve as increasingly detailed descriptions of the task are furnished. The differences between subsequent prompts are highlighted in Table 1 of the supplementary material. In the last iteration of our prompt, we provide a wide range of context, examples, and images to allow the LLM to best utilize all the information available to us.

Table 6 details the predictions of the LLM across prompts for the first table in the UIF framework. Here, we can see that the LLMs predictions are initially verbose when limited information about the task is provided. The predictions start to become more concise when sufficient information about the nature and significance of different slots is provided. This is especially noticeable when we provide the complete example index for an example story titled “Talking myself out of a sale". Following this, the outputs provided by the LLM become much less verbose and start to mirror the example index.

Figures 2, 3, 4, 5 and 6 illustrate the differences in output for Table 2 of the UIF. The hand-crafted slots for this table are shown in Figure 1. It is evident from these images that the LLM initially struggles to predict the correct values for the respective slots. As seen before, the outputs returned by the LLM are verbose, indicative of the vague task description. As the prompts provide greater context about the requirement, the predictions also tend to grow more concise.

In the first set of prompts, the LLM attempts to fill all the slots in the table without considering the possibility that they can remain empty. Starting from prompt 4, when it is suggested that slots can remain unfilled, we see that the LLM can identify the slots for which there is not enough supporting evidence in the story. [Fig. 5]

Physical setting:	close, service, small
Social setting:	sales target
Viewer:	seller
Perspective:	wanted
Agent:	seller
Anomaly Type:	result

	Wanted	*Actual*
Theme:	seller	seller
Goal:	obtain client confidence	make sale
Plan:	remain after close	remain after close
Result:	achieve	block
Side+:		
Side-		block assure

Figure 1: The hand-crafted slots for the story titled “Talking myself out of a sale.” The first table here corresponds to the slots addressed in table 1 of the UIF framework, whereas the second table corresponds to the slots addressed in table 2 of the UIF framework.

An interesting observation to note is that the values proposed by the LLM start off with being very verbose. As the values predicted for Table 1 of the UIF are slowly refined, the values predicted for Table 2 of the UIF also follow suit, demonstrating the remarkable ability of LLMs to adapt, even when limited information is provided. This ability might be especially useful when dealing with scenarios where there is insufficient information present in the story or the anecdote.

Slot in UIF	Actual	Prompt 1	Prompt 2	Prompt 3	Prompt 4	Prompt 5
Physical Setting	Pre-Call, Service-Business, small	Auto Glass shop	Sales presentation, Auto Glass Shop, Small business	Display Ad Dev, Auto Glass, Small	Pre-call, auto glass, small	Close, service, small

Social Setting	Business-partner, seller, married-couple	Business meeting for an ad collaboration	Salesperson and business owner couple	Salesperson, Business Owner couple	Business Owner and	Salesperson-client
Viewer	salesperson	Narrator	Salesperson	Salesperson	Salesperson	salesperson
Perspective	expected	First-person	Assumed	Assumed	Assumed	hoped
Agent	wife	Businessperson and Wife	Business owner's wife	Business Owner's Wife	Wife	salesperson
Anomaly Type	theme	Unexpected decision-maker	Perspective contrast	Unexpected decision-maker	Perspective contrast vs. actual	result

Table 6: Responses of the LLM compared across the five prompts used in our experiment. The actual hand-written slots for the index are provided in the second column. Extra information available to the LLM reduces the vagueness of generated outputs, this aligns the output closer to the expected slot values.

Slot Name	Expected	Actual
Theme	Business owner handles all decisions	Business owner's wife has significant decision-making power
Goal	Finalize the advertisement with the business owner	Gain approval from the wife for the advertisement costs
Plan	Discuss and work on the ad directly with the business owner	Business owner and narrator work on the ad, but the wife approves it
Result	Business owner approves and pays for the advertisement	Wife approves the expenditure and issues the payment
Side+	Direct negotiation and completion with the business owner	Inclusive approach involving both the business owner and his wife leads to successful transaction
Side-	Simple transaction with a single decision-maker	Potential for misunderstanding the roles of individuals present, risk of overlooking the wife's authority

Figure 2: Slots generated from prompt 1

Slot Name	Expected	Actual
Theme	Business discussion	Inclusive decision-making
Goal	Close deal with man	Engage with both partners
Plan	Present ad to man	Present ad to couple
Result	Man approves ad	Woman controls finances and approves ad
Side+	Direct sale process	Enhanced relationship and respect
Side-	Ignoring non-obvious influencers	Missing out on crucial decision-making dynamics

Figure 3: Slots generated from prompt 2

Slot Name	Expected	Actual
Theme	Sales interaction	Sales interaction
Goal	Influence business owner	Influence actual decision-maker
Plan	Present ad to the man	Present ad to both, but approval by the wife
Result	Man approves ad	Wife approves ad and payment
Side+	(none specified)	Sale made due to respecting all present
Side-	Potential loss of sale	(none specified)

Figure 4: Slots generated from prompt 3

Slot Name	Expected	Actual
Theme	Housewife	Business-partner
Goal	Hospitality	Approve the ad spend
Plan	Small talk	Review and approve
Result	Hospitality achieved	Approved budget
Side+		Sale made
Side-	Ignored by salesperson	

Figure 5: Slots generated from prompt 4

Slot Name	Expected	Actual
Theme	salesperson	salesperson
Goal	obtain client confidence	make sale
Plan	remain after sale	remain after sale
Result	confidence achieved	fail to make sale
Side+	-	-
Side-	-	fail to obtain confidence

Figure 6: Slots generated from prompt 5

Future Work

Our current experiments are limited by the number of stories available with full, expert-annotated indices. The ideal next step is to test this method on a bigger collection of stories to verify the performance of the generated indices, which we propose be done in either of two ways.

The direct way of verifying the accuracy of generated indices would be to use these indices in a working story explainer and then compare the performance of that system when using expert indices vs generated indices.

We also propose a method of quantitatively measuring the accuracy of the generated indices. This would be a scoring system that matches the indices generated by the LLM to the expert indices, with a numerical score from 5-1 based on how close the match of the index is with the expert in distinct categories for each slot.

With newer and more powerful LLMs being developed, we expect to also see accuracy improve as their reasoning capabilities and knowledge base grows.

References

Burke, Robin D. 1993. "Representation; Storage, and Retrieval of Tutorial Stories in a Social Simulation." PhD Dissertation. Northwestern University

Burke, Robin; and Alex Kass. 1994. "Refining the Universal Indexing Frame to support retrieval of tutorial stories." In *Proceedings AAAF94 Workshop on Indexing and Reuse in Multimedia Systems*.

Schank, Roger; Brand, Matthew; Burke, Robin; Domeshek, Eric; Edelson, Daniel; Ferguson, William; Freed, Michael; Jona, Menachem; Krulwich, Bruce; Ohmaye, Enio; Osgood, Richard; and Pryor, Louise. 1990. Towards a General Content Theory of Indices. In *Proceedings of the 1990 AAAI Spring Symposium on Case-Based Reasoning*. AAAI Press, Menlo Park

Schank, Roger C.; and R. Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.