Generating Chunks for Cognitive Architectures

Goonmeet Bajaj[1*], Kate Pearce[2*], Sean Kennedy[2], Othalia Larue[3], Alexander Hough[2], Jayde King[2], Christopher Myers[2], Srinivasan Parthasarathy[1]

[1]The Ohio State University
[2]Air Force Research Laboratory
[3]Parallax Advanced Research
bajaj.32@osu.edu, kate.g.pearce@gmail.com, sean.kennedy.10@us.af.mil, othalia.larue@parallaxresearch.org, {alexander.hough.1, jayde.king, christopher.myers.29}@us.af.mil, srini@cse.ohio-state.edu

Abstract

Knowledge engineering is an important task for creating and maintaining a knowledge base for cognitive models. It involves acquiring, representing, and organizing knowledge in a form that computers can use to make decisions and solve problems. However, this process can be a bottleneck for designing and using cognitive models. Knowledge engineering is a time-consuming and resource-intensive task that requires subject matter experts to provide information about a domain. In addition, models can acquire knowledge but require significant mechanisms to structure that information in a structured format appropriate for general use. Given the knowledge engineering bottleneck, we propose a solution that relies on natural language processing to extract key entities, relationships, and attributes to automatically generate chunks encoded as *triples* or *chunks* from unstructured text. Once generated, the knowledge can be used to create or add to a knowledge base within cognitive architectures to reduce knowledge engineering and task-specific models.

1 Introduction

Cognitive architectures are a type of intelligent system designed to model human cognition. One such type of system is ACT-R (Anderson and Lebiere 2014): a modular cognitive architecture that includes perceptual, motor, and declarative memory modules integrated with a procedural module that consists of *productions* to represent knowledge about conducting specific tasks. Buffers connect modules in ACT-R (except for the procedural memory), and the contents of buffers at a given moment in time represent the state of ACT-R. ACT-R behavior is guided by a pattern matcher that finds the production(s) best matching the state of the buffers (environment) and the series of executed productions and state changes represent human cognition.

Models developed using the ACT-R architecture can capture various cognitive activities and reproduce aspects of human data such as learning, errors, and patterns of brain activities. However, a significant effort is required to develop and maintain procedural (in the form of productions) and declarative knowledge for each specific task. Declarative memory is an essential component of ACT-R to accumulate knowledge chunks and retrieve chunks. Procedural knowledge often operates over declarative knowledge to complete tasks. Given the separation of these modules, knowledge engineering is an important task for creating and maintaining a knowledge base for cognitive models that involves acquiring, representing, and organizing knowledge in a form that intelligent systems can use to make decisions and solve problems. One reason for this is the difficulty in acquiring knowledge in a structured format that works with productions designed to complete a specific task. However, developing and maintaining both declarative and procedural knowledge leads to a *knowledge engineering bottleneck* for designing and using cognitive models.

Knowledge acquisition is a time-consuming and resource-intensive task that requires subject matter experts to provide information about a domain. Therefore, we propose using artificial intelligence (AI) methods to generate chunks for cognitive architectures to develop the knowledge base required for specific tasks. Recent advances in AI, such as large language models, have significantly improved parsing unstructured text to structured representations (Drozdov et al. 2022; Kirk et al. 2022). Inspired by these advances, we focus on knowledge acquisition to build up the declarative knowledge for models implemented in ACT-R. As a first attempt, we focus on generating knowledge chunks for hybrid cognitive models designed to interact and complete analogical reasoning tasks.

Analogical reasoning is a canonical task in human cognition, and more recently in artificial intelligence, to understand similarities across two or more situations to reason about target situations (Gentner and Smith 2013; Hope et al. 2017). Specifically, models designed to complete analogical reasoning tasks aim to understand knowledge and infer new information by comparing structured representations (Gentner and Forbus 2011; Hough et al. 2023) (see Section 2 for details). For these types of tasks, collecting knowledge in the form of structured representations is a significant bottleneck for cognitive models. Therefore, we propose automatically generating knowledge in *triples* or *chunks* from unstructured text to help facilitate analogical reasoning capabilities in cognitive architectures. The work relies on a natural language processing pipeline that first extracts key entities, relations, and attributes from the input text. Once these key elements have been identified, the system will generate

[*]Work done during an internship at AFRL

triples or chunks that capture the most important information from the text in a structured format using a large language model (LLM). Next, we determine the plausibility of the generated chunks to ensure correctness. Once generated and verified, the chunks generated by the pipeline can be used as declarative memory to bootstrap the analogical reasoning capabilities for models implemented in ACT-R.

In Section 2, we provide more details on the current analogical reasoning capabilities of ACT-R and highlight limitations in the current approach. Section 4 introduces our solution to address the limitations. Finally, in Section 6, we present future directions and a summary of our findings.

2 Analogical Reasoning Capabilities for ACT-R

As mentioned, analogical reasoning is a canonical task that allows individuals to use prior knowledge and experiences and apply them to new tasks. Analogies are defined by comparing the (labeled) relationships between entities and other relations. A popular form of analogy involves mapping or comparing a familiar *source* domain to a less familiar *target* domain where higher-order relations constrain the lower-order relations and guide the analogical mapping process (Gentner and Forbus 2011; Gentner and Smith 2013). Analogies are typically decomposed into multiple subprocesses (Gentner and Forbus 2011): *retrieval*, *mapping*, *abstraction*, and *representation*. The *retrieval* process finds an analog similar to the situation at hand. Next, the *mapping* process considers the two situations and structurally aligns them to generate candidate inferences with a structural evaluation score that provides a numerical measure of how well the base and target align. The *abstraction* process stores the results of comparison as an abstraction and produces a schema or other rule-like structure. Finally, given a partial match, the *representation* process allows for improving one or both analogs to improve the match.

Structure-mapping Engine (i.e., SME; (Falkenhainer, Forbus, and Gentner 1989; Forbus et al. 2017)) is a computational implementation of Structure-mapping Theory (i.e., SMT; (Gentner 1983)). The theory suggests mapping new (i.e., target) and existing (i.e., base) knowledge structures underlies experiential learning. While mapping, one assumes relations in the base also exist in the target. There is a preference for relations over attributes (e.g., and interrelated or second-order relations over lower-order). This preference for greater coherence is referred to as the *systematicity* principle, and it guides the mapping process. In SMT, mappings are restricted to one-to-one correspondence (i.e., one item in base maps to only one in target) and are structurally consistent (i.e., if second-order relations map, then their first-order must too). SME incorporates these features, generates matches between objects and relations (i.e., match hypotheses), and calculates structural evaluation scores. Some previous cognitive models have mapped objects and relations in two domains based on chaining first-order relations using direct matching rather than structure or semantics (e.g., path-mapping (Salvucci and Anderson 2001)). Models using the SME typically generate candidate inferences with deep structure, compared with other models that chain together first-order relations and require a set of rules to map objects in one domain to another. More recently, Hough (Hough et al. 2023) leveraged both the SME and the path mapping model implemented in ACT-R to understand analogical reasoning capabilities within cognitive models.

The model implemented by (Hough et al. 2023) operates over knowledge stored as chunks in declarative memory. It is presented with a base subject for each analogy task, and the pre-designed productions compile the relevant chunks for that subject. The SME aggregates information in two formats: 1) *dgroup* to describe and represent the system as a list of entities and predicates and 2) a vocabulary file. Each *target* subject is mapped to the base subject one at a time. After a target subject is compiled into a representation, the mapping production passes both the complied base and target representations to the SME. SME computes the match hypotheses, a structural evaluation score, and candidate inferences that extrapolate information from the base to the target.

The model can successfully map target subjects to the base subject, and adding the SME module allows for a new method of learning by experience by abstractly comparing new knowledge to existing knowledge through analogical reasoning. As an initial proof-of-concept (POC), Hough et al. designed an experiment to solve analogies among sports representations (Hough et al. 2023) (see Figure 1). The POC

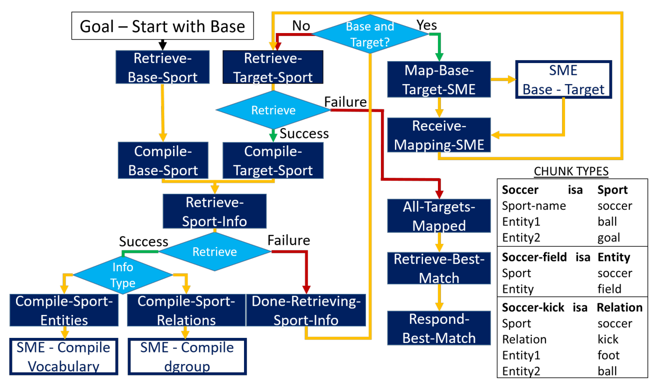

Figure 1: Proof-of-concept demo model processes and chunk types with SME interactions outlined in blue (Hough et al. 2023).

model stored three different types of chunks in declarative memory: 1) sport chunks that contain two entities that "best" represent the sport, 2) single-entity chunks that contain single entities, and 3) relation chunks that contain a relation and entities order to represent their roles with the given relation.

Limitations: For the (POC) model, Hough et al. manually created and designed the knowledge chunks stored in DM for eight sports. The current approach to knowledge engineering is cumbersome, tedious, and time-consuming. Moreover, to reason about other sports, knowledge must manually be formatted and provided to the model, or the model must have a significant amount of mechanisms to parse and format information. Constructing the representations again requires a manual process with the same limitations. Additionally, with the current approach, the model only operates over a fixed set of chunks in DM and is not updated as new information is learned. While it is possible to have the model learn new information and add it to DM, this capability would require providing text or auditory information to the model. In addition, the model would need many new productions to parse text, comprehend it, and format information into triples. This introduces additional bottlenecks and limitations. The current SME methodology for analogical mapping is purely structure-based. For instance, swapping entities across chunks would result in the same analogical reasoning process within the model. For instance, "(*kicks foot ball*)" would map to "(*kicks ball foot*)" and foot and ball would be matched as being similar objects. Whereas, for human reasoning, interchanging the entities across chunks would invalidate chunks and lead to incorrect knowledge representations. Therefore, the similarity score to decide a target could benefit from semantic information about the entities. Specifically, language models from the AI community could serve as a good source of semantic information for these entities.

We chose this cognitive modeling framework because it provides a task-independent capability to reason and develop relations between existing memory that relies on SME to generate similarities rather than task-dependent similarities or functions defined by the modeler (e.g., most previous models in ACT-R). This reduces "tailorability" (Forbus et al. 2017) where model results depend on the representation choices of the modeler. However, the limitations outlined above highlight tailorability and generality are still issues as pre-structured knowledge is provided by the modeler. Given these limitations, we aim to automatically generate the chunks stored in the declarative memory for this POC model to address the knowledge engineering and generality bottleneck. Together, using SME and automatic chunk generation align with best practices to reduce tailborability and increase generality (Falkenhainer 1990). Our approach is outlined in Section 4. In future work, we aim to improve the analogical reasoning approach in ACT-R by including semantic information.

3 Related Work

Current cognitive architectures are not typically equipped with knowledge bases that can capture the world knowledge that humans possess and use (Lieto, Lebiere, and Oltramari 2018). While models implemented in ACT-R are usually equipped with task-specific knowledge, they often lack general cross-domain knowledge. Recent work has explored extending the declarative memory modules of models to enrich the models with general knowledge. Past works have looked at expanding the knowledge layers of ACT-R with ontologies, lexical databases, and knowledge graphs (Oltramari and Lebiere 2012; Lieto, Lebiere, and Oltramari 2018; Salvucci 2014; Ball, Rodgers, and Gluck 2004; Carlson et al. 2010; Bollacker et al. 2008; Speer, Chin, and Havasi 2017; Mintz et al. 2009; Oltramari and Lebiere 2011). Other works have focused on creating knowledge bases for cognitive architectures to use. Existing approaches to creating knowledge bases rely on knowledge graphs that contain information on named entities but lack general world knowledge, use lexical databases with limited relations, and often require large amounts of training data (Carlson et al. 2010; Bollacker et al. 2008; Speer, Chin, and Havasi 2017; Han, Liu, and Sun 2018; Distiawan et al. 2019; Wang et al. 2021).

Algorithm 1: Chunk Generation
**Future work: development of a ranking model*

for $d_n \in D$ **do**
  $E = extractEntities(d_i)$
end for
for $d_n \in D$ **do**
  $T = extractTriples(d_i)$
  $T' = filterTriples(T, E)$
  $T'' = rankTriples(T')^*$
  $T = smeFormatting(T'')$
  return T
end for

To overcome these limitations, more recently, Ribeiro and Forbus populate an ontology with knowledge already defined in an existing knowledge base using a cognitive architecture. Specifically, Riberio and Forbus combine the Companion Cognitive Architecture with the CNLU natural language processing system (Tomai and Forbus 2009) and the SME (Forbus et al. 2017) with the NextKB knowledge base (Forbus and Hinrich 2017). Their system automatically generates training data for analogical training using distant supervision by extracting general knowledge from Wikipedia articles. They combine structural similarity with a language model for word sense disambiguation and fact classification. This work makes excellent strides toward automatic knowledge extraction using several independent modules as a system to extract new knowledge. The authors design an analogical reasoning task to train their system and rely on heavy supervision through ontologies that are not always well-defined for certain concepts, including novel concepts. In Section 4, we introduce our solution that: 1) avoids training an intermediate task to extract knowledge from readily available documents by leveraging a large language model (LLM) to extract triples from unstructured text, and 2) is designed to operate in real-time and maintain a high degree of cognitive plausibility.

4 Proposed Solutions

Given a collection of documents, $D = \{d_1, d_2, ... , d_n\}$, we aim to extract a collection of triples, $T = \{t_1, t_2, ... , t_m\}$ and a set of entities, $E = \{e_1, e_2, ... , e_l\}$ for each document d_n. The set of triples, T, and entities, E, are generated as inputs to the SME. SME requires a ".dgr" file to parse, and the file contains two sections: *entities* and *expressions*. The generated entities, e_l, are formatted as $(defentity\ e_l)$ and serve as a declaration (beginning of the *.dgr* file) of the parameters which will later be used in the expressions describing the relations between entities in the remainder of the *.dgr* file. The generated triples, t_m, are the knowledge chunks or the *expressions* required by SME and are struc-

Sport	# of Generated Triples	# of Filtered Triples By Entities	# of Filtered Triples By Entities & Entity Type	% of Triples Selected based on Entities	% of Triples Selected based on Entities & Type
Archery	220	28	18	0.127	0.082
Badminton	259	125	93	0.483	0.359
Baseball	1073	766	145	0.714	0.135
Basketball	989	251	97	0.254	0.098
Cheerleading	406	132	35	0.325	0.086
Cricket	705	296	159	0.420	0.226
Dance	305	40	23	0.131	0.075
Dodgeball	73	45	16	0.616	0.219
Field Hockey	357	187	131	0.524	0.367
Hockey	183	66	33	0.361	0.180
Ice Hockey	658	285	113	0.433	0.172
Kickball	95	39	25	0.411	0.263
Pickleball	233	107	59	0.459	0.253
Powerlifting	505	147	93	0.291	0.184
Skateboarding	329	55	11	0.167	0.033
Soccer	541	296	136	0.547	0.251
Swimming	181	41	30	0.227	0.166
Table Tennis	252	140	62	0.556	0.246
Tennis	608	336	132	0.553	0.217
Volleyball	450	220	144	0.489	0.320
Water Polo	143	79	43	0.552	0.301
Wrestling	200	63	46	0.315	0.230
Average	**398**	**170**	**75**	**0.407**	**0.203**

Table 1: Generated Triple Statistics

tured as $(relation\ object\ subject)$.

We extract entities and knowledge triples from related Wikipedia articles for each sport. We first extract entities from all Wikipedia articles by parsing each sentence in each d_n, using Spacy's *en_core_web_sm*[1] and extract all nouns as entities to create *entity chunks*, E. Not only are these *entity chunks* used as input to the SME but they are also used to filter relevant triples.

We use the following approach to generate chunks for a specific sport (outline in Algorithm 1). First, we identify the relevant Wikipedia page for each sport to extract information about that sport. Next, we rely on a ChatGPT (GPT-3.5-turbo-16k) based model[2] to extract a triple from each sentence. The model returns a triple, t_m, in the form of ($subject$, $relation$, $object$), a sentiment score, and a confidence score for each source sentence. The collected triples from each sentence in d_n form the set of triples, T for each d_n.

Given the content available on a Wikipedia page, not all extracted triples in T are relevant to describe a sport. Therefore, we rely on filtering and ranking triples to improve the quality of generated chunks. We first filter out triples that do not contain an entity, e_l, in either the subject or object part of the triple. Next, we filter out triples using the Spacy's *en_core_web_sm*[3] to identify *subjects* or *objects* that are named entities (i.e., "ORG", "GPE", "PERSON", "DATE", and "EVENT"). To use the generated and filtered triples with the demo model from Figure 1, the triples are encoded as relations linked to this specific sport in ACT-R (one chunk per relation containing slots for the system (i.e., the sport), the type of relation, and associated parameters of this relation).

Once the triples have been imported into the model's declarative memory:

1. The model retrieves all elements for the current sport (i.e., base or target) one at a time (see Figure 1), and relations are defined and encoded in the *.dgr* file necessary for the SME
2. Once all elements for both the base and target sports have been retrieved and encoded, the model (see Figure 1) calls the structure mapping engine to create mappings and compute similarities which are then assigned to ACT-R chunks.

In our future work, we aim to: 1) filter triples using the extracted relations and relations in existing ontologies or knowledge graphs (i.e., ConceptNet (Speer, Chin, and Havasi 2017)) and 2) develop a ranking model to rank more relevant triples with a higher score to improve the run-time of analogical reasoning with ACT-R by improving the quality of the knowledge chunks available in the DM and reducing the number of non-relevant chunks.

5 Early Results

This section presents our results for *triple generation* and the analogical reasoning task using ACT-R.

[1]https://spacy.io/models

[2]https://huggingface.co/spaces/jingwang/triple_extraction

[3]https://spacy.io/models

Sport	Generated Triples
Badminton	(shuttlecock, not allowed, bounce) *(BWF World Championships, Held in, 1977)*
Baseball	(player, reaches, base) *(Chico Carrasquel, Became, Hispanic All-Star)*
Cricket	(non-players, position, umpire) *(Cricket World Cup, is, International competitions)*
Ice Hockey	(score, tied, shootout) *(IIHF Women's World Championship, Held, 1990)*
Powerlifting	(lifter, lower, bar) *(British Championship, Held, 1966)*
Squash	(winner, receives, point) *(Egypt, Rank, top fifty)*
Table Tennis	(player, hits, ball) *(Li Xiaoxia, Winner, World Cup)*
Volleyball	(player, serves, point) *(Asian Club Championship, Event, women)*
Water Polo	(team, receives, points) *(1904 Olympics, Had, 40 wrestlers)*
Wrestling	(attacker, uses, arm) *(Greece, Participant, World Junior Championships 2004)*

Table 2: Examples of Generate and Filtered Out Triples (italicized) (before SME formatting)

5.1 Results for Triple Generation

We extract triples for 22 sports using their respective Wikipedia Articles: *archery*, *badminton*, *baseball*, *basketball*, *cheerleading*, *cricket*, *dance*, *field hockey*, *hockey*, *ice hockey*, *kickball*, *pickleball*, *powerlifting*, *skateboarding*, *soccer*, *swimming*, *table tennis*, *tennis*, *volleyball*, *water polo*, and *wrestling*. Table 1 contains statistics about the generated triples for each sport and the number of filtered triples. On average, we extract about 398 triples from Wikipedia articles. Next, using the list of entities, E, we can reduce the list of generated triples by about 41%. Finally, by filtering the *subject* and *objects* by their types, approximately 20% of the generated triples remain. Table 2 contains examples of generated triples and triples filtered (in italics) out by the pipeline.

5.2 Results for Analogical Reasoning

Table 3 shows the correspondence identified by the model between badminton and volleyball. For example, the sentences "The server hits the shuttlecock" and "A player stands behind the inline and serves the ball, in an attempt to drive it into the opponent's court" lead to correspondences between *shuttlecock* and *ball*, and *server* and *player*. The model was able to compute meaningful similarity scores (i.e., ACT-R similarity score of 0.08 between cricket and volleyball and 0.25 between badminton and volleyball) and correspondences. However, the model sometimes missed expressions that included several words (e.g., ends when one player wins 4 points ...) or different verb tenses (e.g., hit and hits). From the initial set of results, we see that the language of Wikipedia articles results in a set of relations with high variations, which causes a challenge for the SME. Transformations such as lemmatization on the triples pre-processing im-

Badminton	Volleyball
shuttlecock	ball
bounce	net
(NOT_ALLOWED shuttlecock bounce)	(NOT_ALLOWED ball net)
tennis	newcomb_ball
ends_when_one_player_wins_4_points_or_wins_two_consecutive_points_at_deuce_points	volleyball
(GAME volleyball)	(GAME newcomb_ball)
server	player
(HIT server shuttlecock)	(HIT player ball)

Table 3: Sample Results from Analogical Reasoning within ACT-R/SME

proved our results, and we believe further transformations could increase the correspondences identified.

5.3 Discussion

Our current approach relies on large language models and their ability to function as parsers for information extraction needs. An alternative to this approach is to prompt the large language models to generate chunks for the task. As an initial attempt, we prompt ChatGPT with the following prompt to extract information about different sports:

> Can you generate knowledge triples about sport X in the following format: (entity, relation, entity)?
> Here are some examples:
> (player, kicks, ball)
> (player, scores, point)

Here are example outputs for *soccer*:

- (goalkeeper, blocks, shot)
- (midfielder, passes, ball)
- (defender, clears, ball)
- (coach, trains, players)

An initial analysis of prompting the model for knowledge facts about sports shows that the generated triples are useful for our task. Additionally, the triples are less noisy as compared to the ones extracted from the Wikipedia articles. However, there are generated triples that are not relevant to the task. Here are example irrelevant triples for *soccer*:

- (fans, support, their team)
- (injury time, added to, match)
- (injury prevention, includes, warm-up)

Overall, prompting the model for knowledge about sports also works well for our task.

In summary, we explored both prompting large language models for knowledge and using them as a parser to extract knowledge from unstructured text. We find that these two methods are well-suited for the task. However, it might be beneficial to select one approach over another. For particular domains, using a large language model to parse the domain-specific text might be a better option as the large language model might not contain enough specific knowledge about the domain. For more common domains (i.e., sports), prompting these models for structured knowledge might be promising as the generated triples are less noisy.

6 Conclusion and Future Directions

Knowledge engineering is an important task that remains a bottleneck for creating and maintaining knowledge bases for cognitive architectures. We define a solution to automatically generate knowledge chunks for an analogical reasoning task in a version of ACT-R augmented with an SME module. Analogical reasoning is a crucial aspect of human cognition that facilitates learning, problem-solving, creative thinking, and generalizability. It is also used to benchmark the development of artificial intelligence, particularly in areas such as machine learning and natural language processing and understanding. Our proposed solution uses natural language processing to extract entities, relationships, and attributes to automatically generate chunks encoded as *triples* or *chunks* from unstructured text. As a demonstration, we provide automatically generated chunks to an analogical reasoning model implemented in ACT-R and show that it can use these chunks to determine correspondences between sports through mapping. Our results show a promising first step that reduces the manual effort needed to create knowledge chunks and addresses the tailorability issue to increase generality.

In future work, we aim to: 1) prune the Wikipedia articles to only parse relevant sentences for triples, 2) filter triples based on relations by comparing the relations to existing ontologies and knowledge graphs can also reduce irrelevant triples, 3) train a custom ranking model to select more relevant triples, 4) rely on ACT-R's memory mechanism to prune and query the information source according to the entities of interest to the ACT-R model in the current context (e.g., learn about how volleyball is played but leave out historical information as the model is attempting to start a game), and 5) improve analogical mapping performance within ACT-R by leveraging word embeddings to provide word meaning/similarity to facilitate matching between the *source* and *target* domains.

Acknowledgements

This research was supported in part by Goonmeet Bajaj's appointment as a Repperger Intern to the Department of Defense (DOD) Research Participation Program administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and the DOD. ORISE is managed by ORAU under DOE contract number DE-SC0014664. This research was also partly supported by Kate Pearce's appointment as a Center for Excellence in Education Intern at the U.S. Air Force Research Laboratory, 711th Human Performance Wing, Cognitive Science, Models, and Agents Branch, administered by the Defense STEM Education Consortium. Contents have been reviewed and approved for public release, case number: AFRL-2023-4098. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries or employees.

References

Anderson, J. R.; and Lebiere, C. J. 2014. *The atomic components of thought*. Psychology Press.

Ball, J.; Rodgers, S.; and Gluck, K. 2004. Integrating ACT-R and Cyc in a large-scale model of language comprehension for use in intelligent agents. In *AAAI workshop*, 19–25.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.; and Mitchell, T. 2010. Toward an architecture for neverending language learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, 1306–1313.

Distiawan, B.; Weikum, G.; Qi, J.; and Zhang, R. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 229–240.

Drozdov, A.; Schärli, N.; Akyürek, E.; Scales, N.; Song, X.; Chen, X.; Bousquet, O.; and Zhou, D. 2022. Compositional semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.

Falkenhainer, B. 1990. *A unified approach to explanation and theory formation*, 157–196. Morgan Kaufmann.

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1): 1–63.

Forbus, K. D.; Ferguson, R. W.; Lovett, A.; and Gentner, D. 2017. Extending SME to handle large-scale cognitive modeling. *Cognitive Science*, 41(5): 1152–1201.

Forbus, K. D.; and Hinrich, T. 2017. Analogy and relational representations in the companion cognitive architecture. *AI Magazine*, 38(4): 34–42.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2): 155–170.

Gentner, D.; and Forbus, K. D. 2011. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science*, 2(3): 266–276.

Gentner, D.; and Smith, L. 2013. *Analogical Learning and Reasoning*. Oxford Library of Psychology. Oxford University Press. ISBN 9780195376746.

Han, X.; Liu, Z.; and Sun, M. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Hope, T.; Chan, J.; Kittur, A.; and Shahaf, D. 2017. Accelerating innovation through analogy mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 235–243.

Hough, A. R.; Laure, O.; Myers, C.; and Leung, O. 2023. Integrated Cognitive Model Framework for Analogical Reasoning. *International Conference on Cognitive Modeling*.

Kirk, J. R.; Wray, R. E.; Lindes, P.; and Laird, J. E. 2022. Improving language model prompting in support of semi-autonomous task learning. *arXiv preprint arXiv:2209.07636*.

Lieto, A.; Lebiere, C.; and Oltramari, A. 2018. The knowledge level in cognitive architectures: Current limitations and possible developments. *Cognitive Systems Research*, 48: 39–55.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011.

Oltramari, A.; and Lebiere, C. 2011. Extending cognitive architectures with semantic resources. In *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*, 222–231. Springer.

Oltramari, A.; and Lebiere, C. 2012. Pursuing artificial general intelligence by leveraging the knowledge capabilities of act-r. In *International Conference on Artificial General Intelligence*, 199–208. Springer.

Ribeiro, D. N.; and Forbus, K. 2021. Combining analogy with language models for knowledge extraction. In *3rd Conference on Automated Knowledge Base Construction*.

Salvucci, D. 2014. Endowing a cognitive architecture with world knowledge. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Salvucci, D. D.; and Anderson, J. R. 2001. Integrating analogical mapping and general problem solving: the path-mapping theory. *Cognitive Science*, 25(1): 67–110.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Tomai, E.; and Forbus, K. D. 2009. EA NLU: Practical Language Understanding for Cognitive Modeling. In *FLAIRS Conference*.

Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; and Tang, J. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9: 176–194.