Neuro-Symbolic Commonsense Reasoning: A First-Order Logic and Sub-Symbolic Embeddings Framework with Depth-Limited Search

Anonymous ACL submission

Abstract

Commonsense reasoning has long presented a hurdle between conversational agents and their ability to naturally engage with humans in conversation, as the infinitely dimensional nature of a dialogue's topics presents a significant reasoning challenge in the study of Natural Language Understanding (NLU). Such a system must conceivably be able to act as a generalizable system for evaluating and reasoning about commonsense statements, problems, and queries: in this way, the agent can attempt to quantify the reasonability of a given input. We attempt to address this through the integration of an explainable neuro-symbolic system that leverages Logical Tensor Networks (LTNs) and First-Order Logic (FOL), combined with sub-symbolic representations of topics and a dynamic baseline knowledge base (KB), to enable generalized reasoning over an unbounded domain of subjects presented in the form of natural sentences and achieve state of the art (SOTA) performance on Yale's FOLIO v0.0 dataset. Such a system is also capable of learning even during the inference of the model, and this being the case, we present a methodology through which the immense KBs associated with such a system might be efficiently evaluated to enable the timely training, inference, and growth of the KB.

1 Introduction

While there are many challenges in the realm of natural language processing (NLP), the ability to reason, especially within a 'commonsense' capacity, is a challenge that frequently rears its ugly head in even the most robust agents today. Of course, tackling such a challenge in an efficient, explainable, and extensible manner are also challenges in and of themselves, and so all of these must be considered when attempting to approach the following question: "how do we build a system which can formally reason over a natural language query in an explainable & interpretable way, and do it such that the agent can grow over time?"

Though before we can tackle how we address this question in more detail, we must first understand the fundamental concepts around which our question has been structured.

1.1 Commonsense Reasoning

In the case of commonsense reasoning, the problem is specific in nature, but rigorous definitions of the core elements can be difficult: there is considerable debate on what constitutes reasoning to begin with (Engel, 2003), to say nothing of what constitutes commonsense itself, and this can muddy the waters when considering whether or not a given agent is appropriately tackling the problem of commonsense reasoning.

We must therefore begin by defining sets of axioms that, as precisely and accurately as possible, encapsulate the challenge which we are attempting to address. These sets, then, must therefore minimally include both a working definition of reasoning, as well as what can be considered to be commonsense knowledge. For these axioms, we adopt the work put forth by Davis (2023), and consider the following: commonsense reasoning, generally, is the ability to make intuitive judgments based on prior knowledge and experience; within NLP, commonsense reasoning refers to machines' ability to understand implicit meaning beyond literal interpretation, to include inductive, deductive, and abductive reasoning; and the ability to reason shall be considered as the ability to come to an explainable conclusion using formal modes of reasoning.

This provides us with a more narrowly focused problem which can then be expanded on as a set of explicit requirements. More specifically, we postulate that an agent attempting the task of

commonsense reasoning must be able to accomplish several delineated tasks. First, it must be able to maintain a persistent repository of, ideally growable, information or knowledge to be used in the reasoning process. It should also be able to discover implicit relationships between entities, concepts, and other objects, and then explain or otherwise illustrate the usage of reasoning used to come to a conclusion in a human-interpretable method. Finally, it should not be limited to the structure of the inputs presented and the domains of which the inputs cover. I.e., the system must be able to interpret natural language and reason over a number of domains that approaches infinite (Belle, 2020).

1.2 Logic Tensor Network Overview

We seek to address these requirements through a neuro-symbolic approach, as it can provide for the usage of both formal logic and the integration of large knowledge repositories while still maintaining an overall explainability, or interpretability, of its decision-making process (Hamilton et al., 2022). Which is to say that neuro-symbolic integration, as a field, is concerned with the combination of neural learning models and classical/symbolic knowledge mechanisms, which allows for the application of data-driven computational models against highly descriptive symbolic definitions/rules related to information.

A key component to this approach is in the form of a Logic Tensor Network (LTN), which is a deep learning model that integrates an intermediary first-order fuzzy logic to examine data and compute a representation of the truth of a statement based on an existing repository of knowledge over a continuous interval within [0,1]. This first-order logic (FOL) provides a formal reasoning component to our overall system, a key aspect missing in the modern purely-neural networks applied to commonsense understanding benchmarks. In essence, this allows us to take a natural language statement, approximate it into a FOL representation, and then ground this formal reasoning by backing specific concepts within FOL, such as entities, predicates, and quantifications, as different forms of mathematical operations (quantifications) or various kinds of functions (predicates). More notably, since these predicates can be any sort of function that outputs values between 0 and 1, this allows us to ground them with multi-layer perceptron networks (MLPs), and thereby facilitates the LTN learning the representation of these predicates. However, to ground these concepts the network must first have an understanding of how specific words should relate to specific classifications: we must be able to answer questions such as whether a word is an entity or a predicate, and which tensor or MLP would it be associated with. Typically, this knowledge would come from a structured knowledge base (KB) of predicates, but as was shown in Bianchi et al.'s work (2019), this KB can be enhanced through the integration of entity embeddings, or semantic vectors, as constants within the LTN's KB to improve its ability to inference about the relationship between entities through a form of associative learning, hence the original authors coming to define this as a kind of sub-symbolic commonsense.

1.3 Contributions

As a general overview, we consider our primary contributions to be an accurate, explainable, and performant neuro-symbolic agent capable of reasoning over arbitrary natural language queries and statements, an demonstrative application of sub-symbolic embeddings for deductive reasoning against an LTN's knowledge base in the context of unbounded knowledge domains, as well as a practical depth-limited search system for averaging low runtime complexity to enable knowledge base growth without degradation of inference speed or accuracy of aggregated satisfiability. While previous commonsense reasoning neuro-symbolic agents are noted as being limited to a specific domain of knowledge (taxonomy, construction, etc.; see section 2.3 for more detail), our system is capable of learning over an arbitrary number of domains in an efficient and accurate manner. In turn, our work provides a general-purpose commonsense reasoning agent that is performant both in the frozen/pre-trained case, and in cases where realtime/post-training learning is desired.

2 Related Work

First and foremost, we consider the deepest grounded concept within the overall system: that of the logic tensor network used by our agent. This forms the groundwork for our overall solution, without which we would not have a basis for conjoining the symbolic structure of a knowledge base with the predictive and pattern-recognition capabilities of modern neural networks.

2.1 Logic Tensor Networks

An LTN, or Logic Tensor Network (Badreddine et al., 2020), leverages FOL to represent variables, constants, predicates, and quantifications within a linear, or vector, space. These networks take their inspiration from Neural Tensor Networks (Socher & Chen et al., 2013) and have been effective in several logical reasoning tasks in natural contexts, which has driven us to utilize them for the neuro-symbolic component of our wider commonsense reasoner. This act of learning and reasoning about data and abstract information related to data is done through using a differentiable FOL language to marry the data and logic together, termed as Real Logic, or RL (Badreddine, 2020).

This RL is responsible for the conversion of logical FOL statements/formulas into a computational graph. To accomplish this, the LTN uses a process of grounding, denoted as G, where terms of the logical language are associated to a tensor in the Real field of the vector space by mapping them in the following manner:

Constants are grounded as individuals in some space of tensors of any rank, which can either be static datapoints or learnable embeddings, shown with the following form:

$$G(c) \in \bigcup_{n_1 \cdots n_d \in \mathbb{N}^*} \mathbb{R}^{n_1 \times \cdots \times n_d} \quad (1)$$

Variables are grounded as a sequence of constants, in our case being a word embedding associated with a concept, which can be described as:

$$v = (c_n)_{n \in \mathbb{N}^*} \quad (2)$$

Predicates are grounded as multi-layer perceptron networks with either one (unary predicate, a noun) or two (binary predicate, a verb) variable constants worth of inputs, totaling to 300 or 600 respectively. These inputs are then based through a series of 2 hidden layers with a rectified linear unit (RELU) (Nair and Hinton, 2010) activation function, with a final probabilistic output layer using a sigmoid function. This entire action effectively maps a 300ary or 600ary domain of constants/word embedding elements to a real form [0, 1], which is taken as a truthfulness degree of the overall predicate and variable(s) combination.

Finally, quantifications are grounded as aggregations of a list of truthfulness degrees, which can either be the existential quantification expressed as a generalized mean:

$$pMean_{(u_1 \cdots u_n)} = \left(\frac{1}{n} \sum_{i=1}^{n} u_i^p\right)^{\frac{1}{p}} \; p \geq 1 \quad (3)$$

Or the universal quantification expressed as a generalized mean of the deviation with respect to the truth:

$$p\bar{x}Err_{(u_1 \cdots u_n)} = 1 - \left(\frac{1}{n} \sum_{i=1}^{n} (1 - u_i)^p\right)^{\frac{1}{p}} \; p \geq 1 \quad (4)$$

Where u is a list of truthfulness degrees in [0, 1].

Through this process, the LTN is able to minimize the problem of learning the reasonability of a query into determining how well a query satisfies a set of known rules, i.e., for a given formula the network must find groundings of variables, constants, and predicates that maximize the satisfiability of all the quantifications in the KB. As an expressed property of these LTNs, resultant from the method through which it learns and the fuzzy semantics of connectives, they are also able to examine combinations of various quantifications, predicates, constants, and variables that it has not been trained on before, which makes it an ideal choice for us as any given KB will inherently be incomplete when applied to an arbitrary domain or subset of domains from the set of all possible domains.

Since this architecture relies on this gauge of satisfaction, it can also be extended into a number of domains: binary, multi-class, and multi-option classification are all possibilities with regard to its reasoning. This satisfaction is also a soft value, in the sense that one can arbitrarily, dynamically, or statically decide on a decision threshold anywhere within the range of 0 to 1, allowing for something resembling confidence levels in answers, requiring a given level of satisfaction in either a query or within its own knowledge base itself before the LTN would make a decision or determination.

2.2 Applications of LTNs with Embeddings

We have decided to incorporate sub-symbolic embeddings in an attempt to increase our agent's ability to reason about information not present within its KB, but the intuition behind such a decision is not necessarily immediately obvious. To explain this, we turn to the fruitful work of Bianchi et al., who originally explored the concept in their paper "Complementing Logical Reasoning with Sub-Symbolic Commonsense". In this paper, they explore the concept of complementing the entity information within an LTN with large vectors of sub-symbolic information, leveraging

word2vec as their word embeddings of choice. In this research, they demonstrate within the domain of animal kingdoms (and taxonomy in general) that including this information improved the ability of the corresponding network to successfully reason about whether or not relationships between different kinds of entity classes existed. For example, even if the network did not know if a tiger was a feline, it would be able to make the deduction that it was a feline since the embeddings for a cat and a tiger were relatively similar, and the network knew that a cat was a feline.

With this concept in mind, we sought to expand upon the idea by leveraging the same sort of architecture, but applying it more widely to natural language queries in general, rather than targeting a specific domain: a concept which is explicitly mentioned as a potential avenue of future research within their conclusion which we have been happy to investigate. We additionally change our word embeddings to ConceptNet's Numberbatch, in the hopes that its improved performance, increased dimensionality, and decreased latent bias (Speer et al., 2017) would improve the ability of the concept to generalize to the case of natural language queries.

2.3 Limitations of Related Architectures

Within the wider field of commonsense reasoning, and indeed reasoning in general, then, we are presented with several common problems that we wish to address. Consider for a moment the current SOTA systems for logical reasoning on the FOLIO dataset (a hand-crafted dataset meant to provide a benchmark for both logical commonsense reasoning as well as provide FOL conversions of the associated natural language sentences), which are both LLMs (large language models). In the case of the 2nd most performant system, LogiT5 (Luo et al., 2024), the LLM is carefully fine-tuned to attempt a knowledge distillation of a chain-of-thought system, to provide it with a better ability to handle logical reasoning challenges. This results in comparatively good performance, especially compared with models not fine-tuned in this manner that rely solely on prompt engineering, which has become a common method for approaching this sort of problem. However, this still does not fundamentally give the system the ability to formally reason, and in turn the agent preforms worse than the current SOTA (state of the art) system, DetermLR (Sun et al., 2023), which combines the impressive generative and QA powers of GPT-4 with a Neuro-symbolic reasoning agent for iteratively reasoning over premises to attempt to verify a target premise. This results in a system that can effectively reason over a more generalized domain of potential questions, but continues to ultimately rely on a an LLM to reach logical conclusions, which remains a key weakness of LLMs given the mechanism of logit selection and its probabilistic nature.

3 Methodology & Architecture

We seek to address our previously outlined requirements, then, by extending the concept of sub-symbolic commonsense reasoning to more generally cover an unbounded domain of problems within commonsense reasoning (that is, natural language queries and their truthfulness) in an efficient and still-interpretable manner (Figure 1).

Such a design involves being able to resolve said natural language (NL) into approximate FOL statements, and then determine the truthfulness of FOL quantifications that may contain never-before-seen variables, constants, or predicates. In our case, the FOLIO dataset spares our design from needing a more complex semantic parser, although we must instead do translation of raw text into executable code. Additionally, we must consider the method by which we will efficiently search our dynamically expanding KB.

3.1 Sub-Symbolic Reasoning

Within the context of the overall architecture, we can take our extracted variables and predicates, and then resolve these concepts/words into corresponding embeddings. For instance, if we had the phase "All birds fly", we could consider X to be our placeholder for the set of all birds, bird to be a unary predicate (noun), and flying to be a binary predicate (verb). We could then take these 3 components and resolve each of them to a vector embedding for the given word, and use these as the input into our predicates (MLPs).

This results in two different cases, the first being when the word being associated with a given variable is already represented within the Numberbatch embeddings, meaning that it can be directly mapped to this embedding. The other potential case is when the word to be associated with a variable is not within Numberbatch's embeddings, meaning that we must choose a suitable basis vector from when the true

representation can then be learned. In this case, we make use of a learnable variable which passes the constants (vector elements) through a gradient function grad_fn(). The suitable vector, then, will be a word whose concept has a suitable ambiguity to sit within the mean of any other possible given concept: based upon denotative meaning, the vector associated with the word "anything" is used as a generic to learn the unknown word from.

3.2 Tempering Knowledge Base Growth

Addressing this practical issue of efficiency in its growth can thus be done through an application of classical search algorithms, considering that the KB is ultimately a directed and connected graph. Through a use of a depth-limited search (DLS) against explicitly referenced predicates, we can rely upon the ability of the network to find associative relationships between KB entities that would otherwise be obscured through cutoff failures of a DLS, thereby allowing for fast parsing

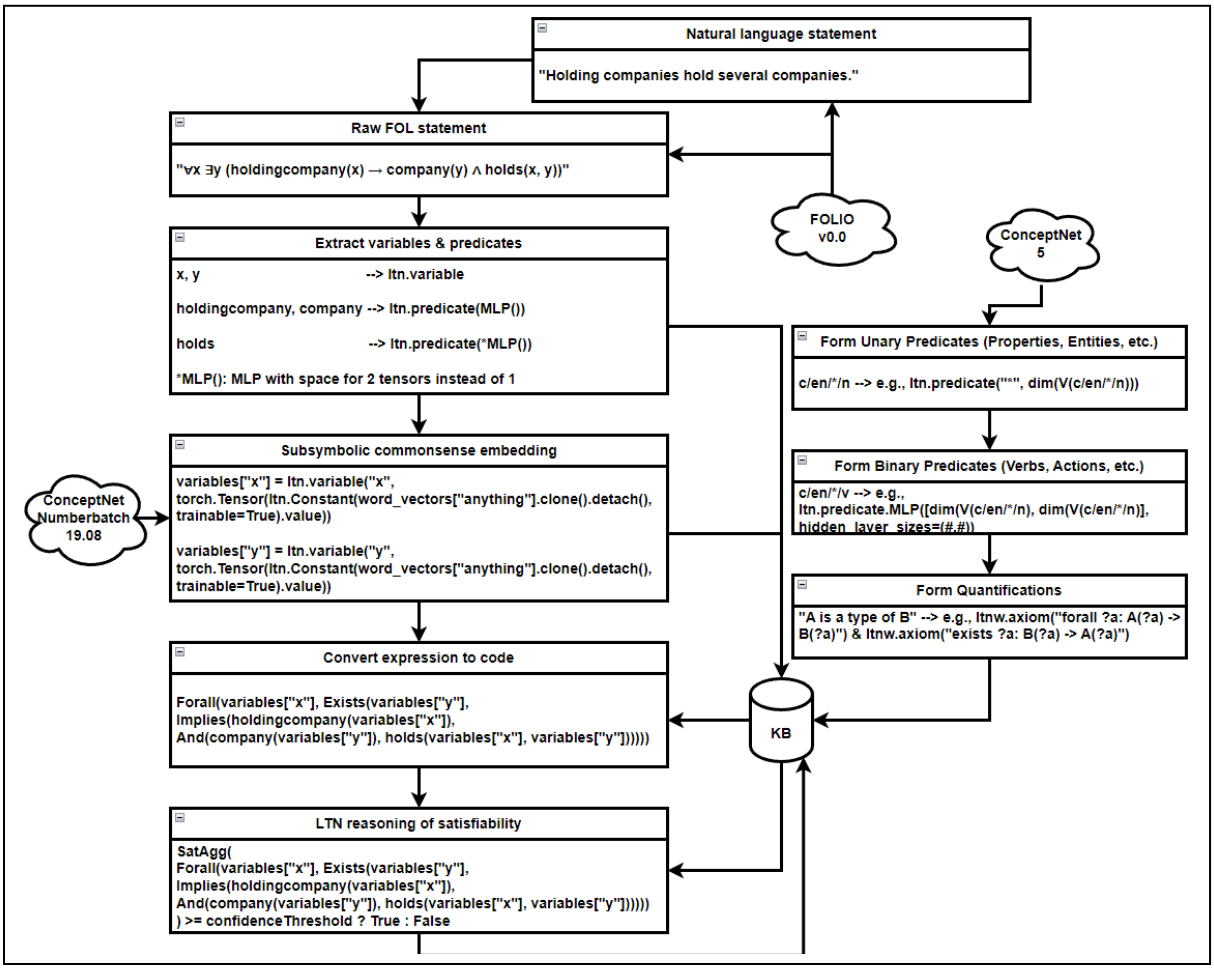

Figure 1 An overview of the system architecture during the inference process

As the task of commonsense reasoning over natural language queries inherently involves an unbound range of domains of knowledge, an expansive knowledge base must be maintained, with embeddings, predicates, quantifications, and all to maintain a suitable amount of reference knowledge for reasoning about unknown topics, entities, relationships, and other considerations. This also needs to be done while accounting for the computation required by the normal inference and learning processes (Figure 1).

of the KB while maintaining a strong degree of relational understanding about the symbolic data and explainable data-driven learning from the LTN itself.

3.3 Training with depth-limited search

Careful consideration must be given to the training component of the system, as a poor approach can result in a non-starter on otherwise solid foundations. In terms of this architecture, we do not do things considerably different from any

other LTN-based agent: we look to the metric of satisfaction as the primary performance determiner. This satisfaction is a representation of how well a given input satisfies the KB of the LTN: for instance, a contradictory or otherwise unreasonable statement would go against most learned quantifications and generate a comparatively low satisfaction score for those quantifications.

These sets of satisfaction can be taken in aggregate, giving us the overall satisfiability of the input, which is our final determiner of reasonability for an input. The resulting metric can then be compared against a confidence threshold, adding an additional hyperparameter that can be turned to impose a restriction on how true or untrue the TLN must evaluate a statement to be before applying a label of true or false.

In the case of the FOLIO dataset, we must also handle written FOL expressions that were created from natural language and evaluate them as code. This process is handled by an abstract syntax tree, which effectively takes the place of a semantic parser that would otherwise be required to attempt the approximation of natural language into an FOL statement.

With the large nature of the KB being used, and with the potential for it to go even larger as the model continues to be trained and make inferences, there becomes a practical consideration for the maximum possible size before training the network becomes unfeasible for the majority of institutions. To address this, we consider a method adopted from classical search algorithms that might considerably enhance the efficiency with which a KB is parsed and in turn substantially increase the maximum possible size of a given LTN's KB.

As a KB in the case of our system is not particularly different from a graph in its overarching architecture, we apply a depth-limited search, a modification of the traditional depth-first search, to our KB when considering a training dataset or an inference query. Essentially, we can reduce the number of quantifications that need to be considered for a given set of inferred premises, predicates, and variables by only computing the aggregate satisfiability of quantifications that are directly associated with the terms being evaluated. More specifically, we evaluate all given terms within a query/statement, and only include entities from our KB that are directly associated with these terms. E.g., if a catfish is mentioned, the KB entity for "fish" will be included, but not the shallower parent of "species".

However, this effectively limits the depth of our KB to 1 with regard to any given term, and so we must address the potential loss of important relational information in more complex queries. In this case, the sub-symbolic knowledge from vector embeddings provides the network with the ability to still attempt to infer any distant relationships between given entities.

4 Commonsense Reasoning Experiments

The driving goal of our following experiment is to illustrate the capabilities of a set of models, and primarily to show the ability for our architecture to do efficient training over an extremely large KB while being able to effectively reason over commonsense knowledge as a function of parsing natural language converted into FOL over a variety of domains of potential topics/referential knowledge. Additionally, we will more generally consider the ability of the network to utilize commonsense and successfully reason over the aforementioned queries.

4.1 Datasets

Perhaps one of the single most important aspects of the research conducted in the manner in which the datasets are arrayed: without a strict adherence to reasonable principles of dataset construction; primarily in the case where training, validation, and testing sets might not be explicitly provided, the entirety of the work, no matter how thorough the experiments, can be undermined. We must then carefully evaluate our utilized language resources and our benchmarks, with a special care for the distribution of samples and testing methodology in the latter.

Concerning the datasets themselves, we consider four separate datasets, with two being used as language resources and two being used as benchmarks for evaluation. These language resources, ConceptNet 5 and ConceptNet's Numberbatch, facilitate the formation of our initial KB and the sub-symbolic reasoning components respectively. On the other hand, our benchmarks are made up of Bianachi et al.'s Animal Kingdom dataset (2019) and Han et al.'s FOLIO dataset (2022), which provides us with both baseline comparative performance as well as a NL-based FOL query benchmark to test our expanded

commonsense reasoning capabilities against, respectively.

The original dataset, identical to the D3 or setting_three dataset in Bianachi et al.'s (2019) paper, hereon referred to as the animal kingdom benchmark, acts as a proof of concept for us to train and test against, and allows for a validation of the new sub-symbolic embedding source (ConceptNet's Numberbatch vs Word2vec in the original work). This benchmark also allows us to conduct an experiment such that we can limit the system to a single general domain (taxonomy) and test our optimization system for both relative gains and deficits when compared to the much slower, but potentially more accurate, unoptimized variant of the architecture

Similarly, FOLIO provides us with an important stopgap in capability: it provides natural language queries/statements that have already been converted by a team of experts into full FOL quantifications. In contrast to the previous benchmark, the FOLIO dataset is a dedicated commonsense reasoning benchmarkt that is meant to test our network's ability to implicitly understand the truthfulness of given natural language statements/premises which are presented in a FOL format.

4.2 Results

We first consider the metrics of both LTN designs from this research, one where the full dataset is applied and one where the depth-limited search is applied to the KB prior to training, in effect creating a subset of the KB of directly relevant quantifications.

Network	LTN_{Full}	$LTN_{Optimized}$
Quantifications in KB	264,313	91,592
Epochs Trained	35	
Time Trained	12 hours	1 hour
Mean sat_agg Score	0.73	**0.75**

Table 1 A comparison of epochs trained and network sat_agg scores between unoptimized and optimized network KBs on the animal kingdom's benchmark

As the above table shows, we see an expected rise in epochs that are able to trained over a period of time through the optimized KB parsing method, and accordingly the mean satisfiability is able to reach a much higher level as the quantifications being evaluated are much more frequently relevant to a given input. This thereby results in the system being able to better handle any given query that is evaluated against the trained term representations (predicate MLPs).

We can also consider the opposite case, where much quicker training can take place when the depth of the KB is limited to only immediately relevant terms. With this speed up in training time, we can establish a direct decrease of the resources (electricity) and computational power (CPUs, memory, GPUs, etc.) that are needed to achieve a given level of satisfiability/convergence. Additionally, a side effect of the KB being used to evaluate the queries seems to be that the mean sat_agg score is able to rise more quickly (and to a higher number by the point of convergence), which we expect as indirect quantifications will not be trained as much (or at all), leaving their sat_agg scores at whatever it is at the point of integration into the KB.

Class	LTN_{Full} F1 Score	$LTN_{Optimized}$ F1 Score	$LTN_{Binachi}$ F1 Score
Animal	0.97	**0.98**	0.88
Fungus	0.15	**0.93**	0.80
Mammal	0.93	**0.98**	0.89
Plant	0.61	**0.98**	0.82
Bacteria	**0.61**	0.55	0.60
Eukaryote	0.99	**1.00**	0.99
Species	**1.00**	**1.00**	**1.00**

Table 2 A baseline F1 performance comparison between the LTN network types and the original LTN from Binachi et al.'s work

With this table, we begin to see a more direct comparison of performance between different forms of LTNs, including Bianachi et al.'s (2019) original system of which we find a deal of inspiration from, especially as it relates to the sub-symbolic component of the system. As is illustrated, the optimized model ends up with leading performance in all metrics aside from the Bacteria class, and is superior in F1 score to all classes when compared to the full model. Despite the full model technically containing more information which may occasionally have relevance, the sub-symbolic has done a good job of maintaining the most important relational information.

More notably, the optimized nature of the system has clearly allowed the LTN to better converge on accurate representations of the term

constants over the same time that the full LTN was given, which is most visible in the Fungus class where the full LTN was unable to find an accurate representation for the associated terms (variables, predicates, etc.) with the limited epochs it was able to train. Comparatively, the optimized LTN was able to train through to convergence within the same timeframe, which is responsible for this enhanced convergence. With a sense of relative performance between the optimized KB parsing network and the full LTN now established, and a confirmation of the ability of the networks to employ sub-symbolic commonsense reasoning in the same domain as their inspirational paper, we can now consider a fuller test where both networks converge on our new benchmark, FOLIO.

Dataset	LTN_{Full} F1 Score	$LTN_{Optimized}$ F1 Score
FOLIO v 0.0	**0.76**	**0.76**

Table 3 A comparison of F1 scores between the unoptimized and optimized network types on the FOLIO v 0.0 benchmark

While the mean satisfiability from the previous table certainly shows that the applied models are far from perfect fits for the benchmark, this is expected due to the complexity of the relationships between various quantifications. On the other hand, the resulting in the F1 scores in the above table provide a direct glimpse into the real-world performance of the networks. Normally, we would expect the performance of the optimized network to suffer: while it is fast, it inherently reduces the available relationships from which conclusions can be drawn. However, the above graph shows that DLS limiting indirect relationships between entites, the sub-symbolic component of the agent is able to compensate for this and maintain the sensitivity and recall of the agent. With this confirmation made, we can now evaluate the absolute accuracy of the agents versus other networks that have been benchmarked against FOLIO, including the current SOTA results and a variety of high-performance LLM-based and Neuro-Symbolic-based strategies.

LTN Full	LTN Optimized	Determ LR GPT-4 + NS	ChatGPT -	RoB-L -	Logi T5 Multi-Task
76.75 %	**77.03 %**	75.49 %	54.41 %	64.71 %	74.02 %

Table 4 Accuracy table of various networks testing against the FOLIO v 0.0 benchmark

Finally, we can compare the overall performance of the network designed in our work against a variety of other networks on the same dataset. As can be noted in the table, our design achieves SOTA performance against other highly performant models like DetermLR, a GPT-4 based solution that incorporates a neuro-symbolic reasoning component (Sun et al., 2023), and LogiT5 (Multi-Task), which is a Flan-T5 seq2seq model trained on multi-task learning via LogiGLUE's in-domain data (Luo et al., 2023). Naturally, these results are highly encouraging and seem to mirror conclusions found in the respective papers that neuro-symbolic reasoning agents present a strong method by which complex natural language queries can be queried in an explainable manner. There is also the note that the LTN utilizing the full KB saw a slightly lower accuracy than the optimized LTN, which may be due to the previous thought that a larger KB can, in some cases, result in a degraded confidence (mean sat_agg score) in the network which impairs its ability to reach a correct conclusion.

5 Conclusion

With all said, then, we have shown that the combination of sub-symbolic word embeddings combined with the neuro-symbolic LTN and FOL parsing logical reasoning architecture is entirely capable of handling both commonsense reasoning reasonability classification in the context of approximated natural language queries as well as domain-specific truthfulness classification.

As LTNs are part of a wider concept of neuro-symbolic systems, this also brings with it an explainable and interpretable system which is both simple and intuitive to customize the KB of. Such a design also integrates easily with the concept of embedding vector information into words to supplement, or perhaps completement, existing logical rules, which in turn can be enhanced greatly in the manner of scaling by the integration of a depth-limited search system for optimizing the sections of the overall KB that are utilized for evaluating a given input.

The prospect of continuing the research done here therefore remains an exciting and attractive option to pursue: with additional work it appears more than possible to develop the existing system presented within this paper into a very powerful and highly generalizable commonsense reasoning agent.

5.1 Limitations

While the initial results of our research have shown a great deal of promise when it comes to furthering the concept of the architecture, there is still a deluge of future work ahead to both continue testing the system and to improve upon its capabilities. This future work, then, can be delineated into the subsets of enhancements/feature additions, evaluations, and stress testing.

Enhancements, or feature additions, would include functions that are presently absent but could be addition to bring about additional capability to the system, be that in the context of the existing functionality or opening up the opportunity for entirely different modes of data to be integrated into the agent. Perhaps most considerable of which would be a system for the approximation of natural language sentences into FOL. This would enable true end-to-end commonsense reasoning within a conversational context, but such semantic processing is a major challenge in its own right, albeit one that has seen advancement in more recent times (Arabshahi, 2021). Consolidation of the KB over time is another performance-related matter, as given the dynamic nature of the KB, it is prone to slowly being bloated by obsolete, redundant, or otherwise unhelpful or even detrimental nodes within its graph: consider how isugly() may be a predicate included within the KB from the ConceptNet 5 dataset as an attribute of an object, and how this could be diluted by a later-learned ugly() predicate based off the usage of the word in a sentence.

Similarly, evaluations constitute applying the agent to a variety of different benchmarks and datasets: this is a specific focus on taking the framework, in whatever state that may be, and applying it to obtain experimental results that can be used to gauge the usefulness of the project and potential future applications/work going forward. A clear standard of which is the Winogrande benchmark (Sakaguchi et al., 2019), an expanded upon Winograd benchmark (Levesque, 2012) which has become a focused-upon test by a number of LLMs and GPTs. Testing against this benchmark, as well as related benchmarks is reliant upon the system's ability to translate natural language directly into executable FOL, which remains a limitation.

Stress testing, then, carries with it a more inwardly reflective sense of evaluation, focusing on topics such as determining the limits of the current architecture and what sorts of edges it can or cannot handle, and how well or poorly it handles such cases. While LTNs were shown to be naturally resistant to contradiction during experimentation, and we have done a degree of formal analysis of how an adversarial actor may be able to influence an agent's "ground truth" or KB, there are still many areas for improvement and our implementation is by no means a comprehensive security suite. Notably, the system does not offer a strong guarantee that the information inserted into the network will always be aligned with the user's intentions for the AI: such a system will require a substantially more complex approach, bringing with it ever-present moral dilemmas should such a system be given more authoritative control over what data is considered desirable or not.

Acknowledgments

Withheld per anonymity requirements.

References

Akinkunmi, B. (2018). Towards A First Order Logic Representation of Complex Situations in Natural Language Understanding. In Journal of Science and Logics in ICT Research. University of Ibadan.

Arabshahi, F., Lee, J., Gawarecki, M., Mazaitis, K., Azaria, A., and Mitchell, T. (2021). Conversational Neuro-Symbolic Commonsense Reasoning. The Thirty-Fifth AAAI conference of Artificial Intelligence (AAAI-21), 35(6), pp. 4902-4911. Association for the Advancement of Artificial Intelligence.

Badreddine, S., Garcez, A., Serafini, L., and Spranger, M. (2020). Logic Tensor Networks. In Artificial Intelligence, volume 303. Elsevier.

Belle, V. (2020). Symbolic Logic meets Machine Learning: A Brief Survey in Infinite Domains. International Conference on Scalable Uncertainty Management, pp. 3-16.

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. (2003). A Neural Probabilistic Language Model. In Journal of Machine Learning Research 3, pp. 1137-1155.

Bianchi, F., Palmonari, M., Hitzler, P., and Serafini, L. (2019). Complementing Logical Reasoning with Sub-Symbolic Commonsense. In RuleML +RR 2019: Rules and Reasoning, pp. 161-170. International Joint Conference on Rules and Reasoning.

Davis, E. (2023). Benchmarks for Automated Commonsense Reasoning: A Survey. Association for Computing Machinery Computing Surverys.

Engel, P. (2003). Reasoning and Rationality. In Dictionary of Cognitive Science: Neuroscience, Psychology, Artifical Intelligence, Linguistics, and Philosophy, pp. 315-316, New York, United States, October. Routledge, 1st edition.

Han, S., Schoelkopf, H., Zhao, Y., Qi, Z., Riddell, M., Benson, L., Sun, L., Zubova, E., Qiao, Y., Burtell, M. and Peng, D. (2022). Folio: Natural language reasoning with first-order logic, distributed via arXiv preprint arXiv:2209.00840, 0.0.

Hamilton, K., Nayak, A., Božic, B., and Longo, L. (2022). Is Neuro-Symbolic AI Meeting its Promise in Natural Language Processing? A Structured Review. In Semantic Web Journal.

Han, S. et al. (2022) FOLIO: Natural Language Reasoning with First-Order Logic. ArXiv Preprint.

Harris, Z. (1954). Distributional Structure. In WORD, 10:2-3, pp. 146-162.

Kumar, R. S. S., Nyström, M., Lambert, J., Marshall, A., Goertzel, M., Comissoneru, A., Swann, M., Xia, S. (2020). Adversarial Machine Learning-Industry Perspectives. In IEEE Security and Privacy Workshops (SPW), pp. 69-75, San Francisco, CA, United States, doi: 10.1109/SPW50608.2020.00028.

Kingma, D., and Ba, J. (2015). Adam: A Method for Stochastic Optimization. ICLR (Poster). 3rd International Conference for Learning Representations.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The Winograd Schema Challenge. Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. Association for the Advancement of Artificial Intelligence.

Luo, M., Kumbhar, S., Shen, M., Parmar, M., Varshney, N., Banerjee, P., Aditya, S., Baral, C. (2023). Towards LogiGLUE: A Brief Survey and A Benchmark for Analyzing Logical Reasoning Capabilities of Language Models, distributed via arXiv preprint arXiv: 2310.00836v2.

Nair, V. and Hinton, G. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceddings of the 27th International Conference on Machine Learning, pp. 807-814. ICML.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, in proceedings of AAAI 31, ConceptNet 5.7 & Numberbatch 19.08.

Sakaguchi, K., Bras, R., Bhagavatula, C., and Choi, Y. (2019). WinoGrande: An Adversarial Winograd Schema Challenge at Scale. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), pp. 8732-8740. Association for the Advancement of Artificial Intelligence.

Singh, S., Wen, N. Hou, Y., Alipoormolabashi, P., Wu, T., Ma, X., and Peng, N. (2021). COM2SENSE: A Commonsense Reasoning Benchmark with Complementary Sentences. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 883-898. Association for Computational Linguistics.

Socher, R., Chen, D., Manning, C., and Ng, A. (2013). Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Advances in Neural Information Processing Systems 26. NeurIPS.

Song, J., Lee, J., Choi, J., and Kim, I. (2020). Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences. In Journal of Computational Design and Engineering, pp. 563-576. Oxford University Press, volume 7, issue 5.

Sun, H., Xu, W., Liu, W., Luan, J., Wang, B., Shang, S., Wen, J., Yan, R. (2023). From Indeterminacy to Determinacy: Augmenting Logical Reasoning Capabilities with Large Language Models, distributed via arXiv preprint arXiv: 2310.18659v1.

Yan, L. Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., Gašević, D. (2023). Practical and ethical challenges of large language models in education: A systematic scoping review. In British Journal of Educational Technology. British Educational Research Assoication.