Shaped-Charge Architecture for Neuro-Symbolic Systems

Boris Galitsky
Knowledge-Trail Inc. San Jose CA US
bgalitsky@hotmail.com

Abstract

In spite of the great progress of large language models (LLMs) in recent years, there is a popular belief that their limitations need to be addressed "from outside", by building hybrid neurosymbolic systems which add robustness, explainability, perplexity and verification done at a symbolic level. We propose shape-charged learning in the form of *Meta-learning/DNN - kNN* that enables the above features by integrating LMM with explainable nearest neighbor learning (kNN) to form the object-level, having deductive reasoning - based metalevel control learning processes, performing validation and correction of predictions in a way that is more interpretable by humans.

Introduction

Large Language Models (LLMs) are at the forefront of the AI revolution, transforming how we interact with technology and the world around us. However, as with any technology, LLMs come with their own set of limitations, understanding which is crucial for the continued development and refinement of these models, ensuring they can be used safely and effectively.

These limitations can range from issues with understanding context to generating misinformation, ethical concerns and a lack of creativity. They represent fundamental challenges in natural language processing (NLP) and ML, and addressing them is crucial for the safe and effective use of LLMs. These limitations also reflect broader concerns in the field of AI, including the spread of misinformation, ethical implications, and the quest for genuine creativity.

Numerous strategies can be employed to address these limitations, encompassing robust evaluation methodologies based on human assessment, automated metrics, adversarial and out-of-distribution testing. Effective prompt design, such as prompt engineering, prompt-based learning, meta-prompting, and meta-learning, as well as prompt-based fine-tuning, plays a crucial role in enhancing the accuracy and utility of model outputs. Researchers are actively exploring techniques like differential privacy, fairness-aware machine learning, and explainable AI methods such as LIME and SHAP (Safjan, 2023).

One of the pitfalls associated with neural AI involves the risk of misinterpreting the capabilities of LLMs and overestimating their abilities. A key aspect of the human mind, known as "theory of mind," involves the ability to reason about the intentions, beliefs, and knowledge of entities in the real world, including oneself (introspection) (Otto and Tuedio, 1998).

When applied to an LLM, this concept implies the expectation that a model trained to generate captions for images understands the content of the pictures and the captions it produces. The term "understanding" in this context refers to the mapping of the real world into a symbolic representation. Users of LLMs may be surprised when deviations from the types of images in the training dataset result in the generation of nonsensical text by the LLM.

We construct a hybrid ML framework, initiating with neural gradual descent and culminating in nearest neighbor learning (Galitsky et al., 2023). This approach seeks to amalgamate the advantages of both realms: the efficiency of neural learning, coupled with the explainability and meaningfulness of nearest neighbor learning, and the precision of reasoning, which validates and rectifies the results produced by LMMs. In the prediction process, we employ neural gradual descent initially, and upon obtaining a candidate prediction, we apply nearest neighbors to produce a precise, interpretable outcome. Consequently, the prediction sessions of the shaped-charge ML 'explode' upon reaching the final stage, following the application of a set or sequence of relevant samples.

Our approach addresses various limitations associated with standalone deep neural networks (DNNs), including the lack of explainability, explicit generalization with common sense, vulnerability to adversarial attacks, and a deficiency in introspection. To overcome these limitations, we integrate an explainable ML component such as k-nearest neighbor (kNN) around an LLM, and employ meta-learning in the form of reasoning about the LMM+kNN learning process.

One implementation of shaped-charge ML is in question-answering, where a neural ML subsystem is followed by a syntactic match with the candidate answer, ensuring verification and correction when necessary (Galitsky, 2023).

A similar approach is applied to summarization, where a kNN verifies the presence of a similar sentence in the text for each summary phrase. Additionally, a transformer-based content generation architecture is included, verified and corrected by a web mining fact-checking component.

Combining symbolic reasoning and numeric processing is known to be a promising direction in AI and applications. Our hybrid architecture can be expressed as *Meta-learning/DNN → kNN: kNN* follows *DNN,* and learning configuration is controlled in metalanguage of inductive learning, such as inductive logic programing (ILP) or discourse analysis as a meta-language for object-level semantics. One way to look at this research is building a discourse theory around DNN learning as if all learning occurs in a text. For other kinds of data such as images, we consider discourse in a broader context at an abstract meta-level.

Hence, shaped charge learning functions as follows: a kNN completes an operational DNN, and meta-reasoning controls them both by verifying and correcting. DNN models and kNN cases are domain specific but meta-reasoning is not. Meta-reasoning controls active learning, the current dataset, generalizes cases of failure at the object level, decides which ontologies to involve, and also controls how the whole ML agent communicates with its peers. Explanation chain rules are also parts of the meta level, along with rules extracted by an inductive learning system such as Inductive Logic Programming.

In a shaped charge, the explosive is formed into an inverted cone with an angle of about 50° (Fig. 1). The base of the cone is oriented towards the front of the projectile. A thin layer of metal forms inside of the cone. At detonation time, the metal shape collapses onto the axis, producing a jet of the liner that expels at a high velocity out of the cavity. The jet stream extends, accelerates, and finally breaks up. A slug formed from the rest of the liner material follows the jet. A flying projectile is associated with an exploration of space by a DNN, and an actual explosion – by finding an exact prediction by an interpretable kNN or symbolic reasoning.

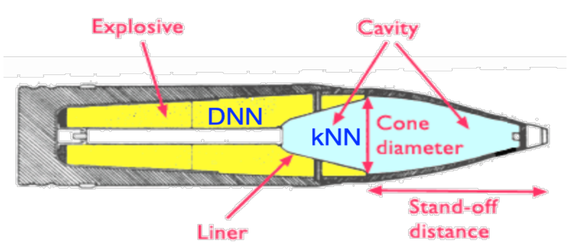

Figure1: An idea of shaped-charge projectile and its realization as an ML architecture

Related Work

An increasing number of researchers are focusing on convergence of neural systems and symbolic systems, with the aim of realizing the third wave of AI known as neural-symbolic learning systems (Mao et al. 2019; Marra et al. 2020). (Kahneman 2011) underscored the importance of a "intuitive, rapid, unconscious, nonlinguistic, and habitual aspects" transformation into "deliberative, logical, sequential, conscious, linguistic, algorithmic, planning-related, and reasoning-related facets": all are associated with deep learning. Many cognitive scientists have explicitly advocated the concept of dual processes corresponding to these contrasting systems (Honavar, 1995), emphasizing the need to integrate neural and symbolic systems. By combining these two types of systems within a comprehensive framework, neuro-symbolic learning systems can be developed, providing LLMs with the capability to handle both perception and reasoning tasks (Yu et al. 2023). It is noteworthy that the idea of integrating neural and symbolic systems, initially known as hybrid connectionist-symbolic models, was originally introduced in (Sun & Bookman, 1994).

Inductive Logic Programming (ILP) (Muggleton 1991) is a sound formalization for finding theories from given examples using first-order logic as its language (Nienhuys-Cheng et al. 1997). ILP algorithms use syntactic bias, which forces syntax constraints on hypotheses, such as the number of variables allowed in a clause, and also semantic bias, which reduces the number of hypotheses based on their semantics, such as whether they are irreflexive or functional.

Metarules control syntactic bias used by many ILP approaches (Wang et al. 2014), including *Metagol*. Evans and Grefenstette (2018) proposed Differentiable Inductive Logic Programming (∂ILP), which is an environment for building logic programs from given samples relying on differentiation. The ∂ILP framework reduces an ILP to an optimization process that can be solved by gradient descent. Its differentiability establishes a promising merge of ILP and neural networks to deal with sub-symbolic and noisy data.

A metarule is a higher-order clause which defines the exact form of clauses in the hypothesis space in an ILP, and defines the learning configuration for an arbitrary learner such as a DNN. For instance, the chain metarule is of the form $l(A,B) \leftarrow m(A,C),n(C,B)$, where l, m, and n are predicates, A, B, C denote predicate variables, and the result allows for instantiated clauses such as: *final_processing_step(A,B):-reorder(A,C), first_processing_step(C,B).*

Consecutive DNN and kNN Architecture

DNNs trained by a gradient descent algorithm are similar to kernel machines in a mathematical sense. Kernel machines store the data points and leverage them directly for prediction by means of a distance-measure function. This nicely improves the explainability of deep neural network (DNN) weights, as they are essentially a superposition of the training set samples. A DNN structure embeds knowledge of the target function into the kernel. Most ML systems and DNNs in particular learn using certain versions of gradient descent (GD, in Rumelhart et al. 1986). Starting with an initial parameter vector w_0 and a loss function, GD iteratively updates the DNN weights w by subtracting the loss's gradient from them, normalized by the learning rate ε:

$$w_{s+1} = w_s - \epsilon \nabla_w L(w_s)$$

The iterations stop when the gradient is zero and the loss value is optimized. Learning via GD assures that its end result is almost always a kernel machine, and it is an invariant with respect to the number of network layers or neuron connection structures.

Kernel machines that implement GD rely on what we call a path kernel. If a learning rate is minimized, the path kernel between a pair of data samples is the integral of the dot product of the GD at the pair of respective points over the path traveled by the learning parameters:

$$K(x, x') = \int_{c(t)} \nabla_w y(x) \cdot \nabla_w y(x')\, dt$$

where $c(t)$ is the path. Informally, the path kernel indicates the distance between the pair of data points as it varies from iteration to iteration. The lower the distance between the variation for x and x_0 is, the higher the weight of x_0 in predicting y.

A kNN can be expressed via GD as well. A kNN requires feature scaling; however, scaling all of the data to the same range is not sufficient, and when the number of features is large, it is impossible to scale the data manually. In a traditional kNN, the distance between the query object and the j-th object in dataset $d(x_q, x_j)$ could be described as

$$d(X_q, X_j) = \|X_q - X_j\|_2^2.$$

For its feature matrix X, the feature can be scaled using a vector A, thus turning the distance function $d_2(x_q, x_j)$ into

$$d(X_q, X_j) = \|AX_q - AX_j\|_2^2 .$$

$d(\boldsymbol{X}, \boldsymbol{X}_j)$ should be replaced by a more general metric: that is $d_L(\boldsymbol{X}, \boldsymbol{X}_j)$. If $L = A^T A$, then $d_L(\boldsymbol{x}, \boldsymbol{x}_j) = (Ax - Axj)^T (Ax - Axj)$. Since mean square error (MSE) is a function of ŷ and ŷ depends on $\|x - x_j\|^2{}_L$, MSE can be minimized by selecting an optimal value of L. Votes, V_j, can be replaced by W_j as in:

$$W_j = \exp\left(\frac{-\|A\boldsymbol{x} - A\boldsymbol{x}_j\|_2^2}{2\sigma^2}\right).$$

Then the GD of the error function E_A with respect to the matrix A, which is minimized to get an optimal A, can be expressed as

$$\frac{\partial E}{\partial A} = 2A(y_i - \hat{y}_i)\frac{1}{\sum_j W_j}\sum_j (y_j - \hat{y}_j) W_j (\boldsymbol{x} - \boldsymbol{x}_j)(\boldsymbol{x} - \boldsymbol{x}_j)^T$$

The path kernel is intended to measure similarity between examples (Fig. 2). In the 2D training case as the weights travel through a path, the model's GD vectors on the weight plane for x, x_1 and x_2 are updated. The kernel $K(x, x1)$ and $K(x, x2)$ are then the integral of the dot product of the gradients $\nabla_W y(x)$ and $\nabla_W y(x_2)$ traveling through the path. Since $\nabla_W y(x) \cdot \nabla_W y(x_1)$ is greater than $\nabla_W y(x) \cdot \nabla_W y(x_2)$ on average traveling through the weight path, y_1 is more important than $y2$ in predicting y, all things being equal.

We visualize the gradient descent in the linguistic space as getting closer and closer to the linguistic expression that makes it complete and truthful (Fig. 3). The question "gradually descends" towards the answer so that the similarity between them increases. We start with the sentence in a certain vicinity of the question, but not close enough: the verb is somewhat similar, but the entity is different. We then navigate through other incorrect entities until we converge on entity = *Lermontov* and predicate = *killed*. The role of verb (x_1) and noun (x_2) in matching a question by its answer is different. As we navigate the space of embedding of a DNN, the facts are not necessarily true, but they support a monotonic convergence. Once we transition to kNN space, it is an abrupt change as we now navigate through factually correct answers but not as smoothly organized as under a DNN.

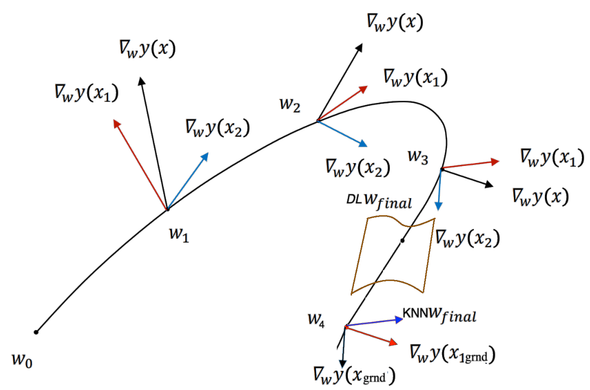

Figure 2: A path kernel first for a DNN and then for a kNN

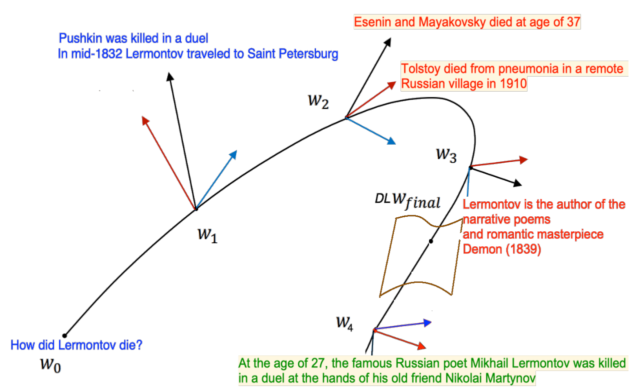

Figure 3: Linguistic interpretation of a path kernel in finding an answer to a question

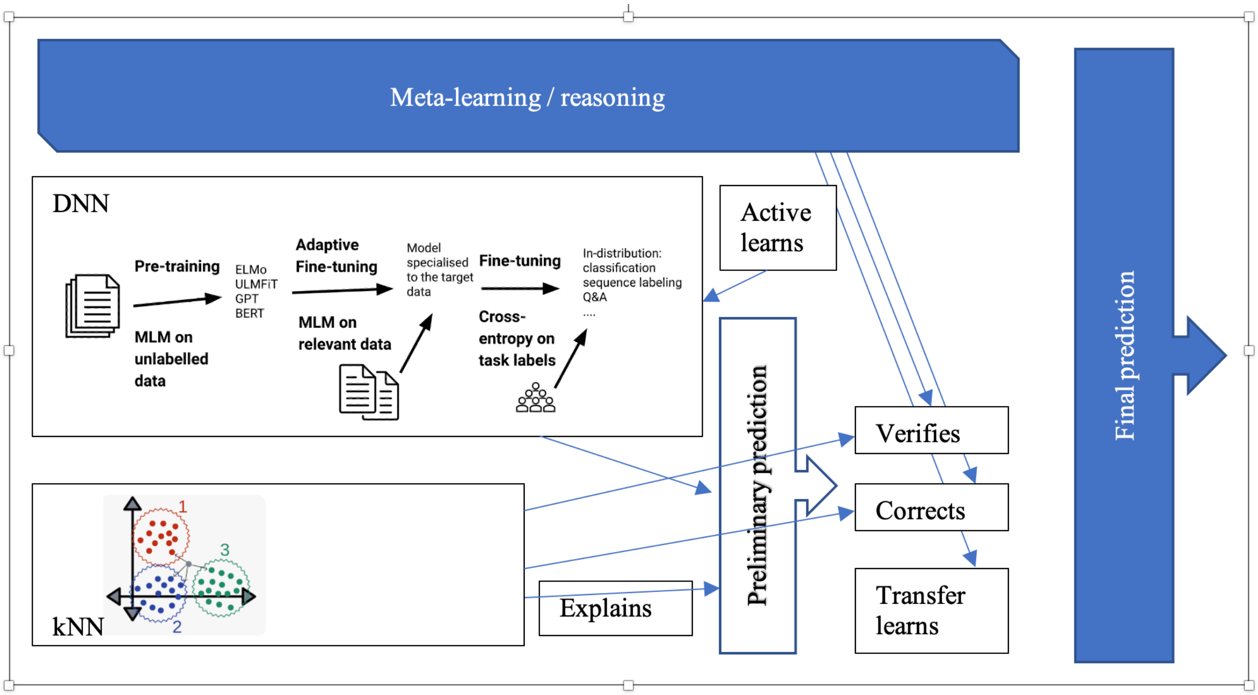

Figure 4: Proposed shaped charge learning architecture simulates maximum entropy production

We finally approach the overall architecture for wrapping an LLM. At training time, training datasets are selected and maintained by a meta-learning component. Based on the current prediction results, meta-learning decides which additional data samples to involve (active learning). From the discourse structure of sample texts, meta-learning automatically discovers how to combine individual DNN processing components. Meta-learning also transfers the trained model to a similar problem domain, if its evaluation is successful in it (Fig. 4).

In the inference time, a prediction made by a DNN is confirmed or rejected by a kNN component. Meta-learning verifies the prediction made by the DNN by maintaining a current set for a kNN and maintaining a threshold. Even if the current prediction is not modified by the kNN, the identified neighbor is used for an explanation of why the input belongs to a certain class. The final prediction is computed as meta-learning assesses the confidence level, given the results of the DNN and kNN.

Applications and Evaluation

In this section, we enumerate application areas in NLP where shaped-charge learning is implemented, analyze the peculiarities of the architecture in each domain, and observe the specific of LLM, kNN and meta-learning used in each application area.

Answering Questions

For question answering, we follow the shaped charge architecture, combining an LLM and direct syntactic/semantic similarity approaches. It is expected to be beneficial since these approaches rely on different but complementary feature spaces. A stand-alone DNN fails answering questions due to a lack of an online phrase structure similar to the one available in the training set, or an inability to generalize to cover various cases. In case of an LLM, it frequently “hallucinates” when there is a lack of a similar enough example in the training set.

To apply a kNN to question answering, semantic parsing is applied to map a natural language sentence to a semantic representation. One such semantic representation is called an Abstract Meaning Representation (AMR), which is represented as a rooted, directed, acyclic graph with labels on edges (relations) and leaves (entities). Leveraging an efficient AMR, this work implemented an instance of a kNN approach, which is termed a “generalization of AMR” (AMRG).

(Galitsky 2020) proposed an architecture for what is now called RAG (Retrieval augmented generation) which is based on a vector database which is a specific implementation of a kNN. RAG architectures are usually not as linguistically deep as AMRG but more efficient in handling large quantities of data.

We observed a very modest $1.89 \pm 0.12\%$ improvement in the performance due to the DNN + AMRG architecture (Table 1). The AMR-only row is a useful case for our ablation study.

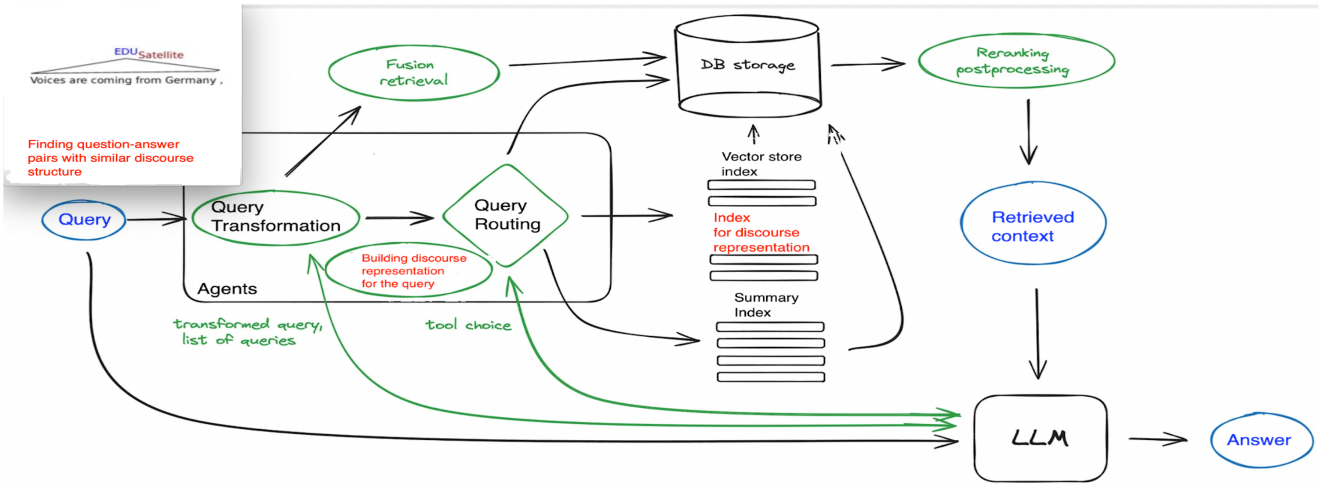

Figure 5: RAG + discourse analysis for answering complex questions

Method	P	R	F1
AMR only	83.3	84.1	83.7
DNN + AMRG correcting DNN errors	92.7	92.3	92.6
DNN vs AMRG as selected by meta-learning	83.1	85.0	84.0
AMRG errors identification	76.8	77.2	77.0
AMRG errors correction	70.2	72.1	71.1

Table 1: Performance of a hybrid DNN + AMRG system on SQuAD 2.0

Leveraging Discourse Structure of Documents Answering Questions

Answering complex questions against long documents with various structure and text genres, discourse analysis is beneficial, especially for specific domains. For example, medical discourse sheds a light on the communication structure of patient-doctor and other communication scenarios in healthcare and should be leveraged to facilitate and automate this communication when possible and practical. We propose a unified framework to represent communication discourse at the meta-level, where the subject of the communication is expressed in a language object.

A RAG architecture leveraging discourse analysis is shown in Fig. 5. We build a discourse representation of a long, complex query. We also build a hierarchical representation of chunks of documents in the form of discourse tree as an addition index, along with a vector store index and summary index. This approach follows the LLM-kNN paradigm of shaped-charge learning.

We compare our approach with four competitive approaches for a long document QA:

(1) ETC (Ainslie et al., 2020) applies global-local attention mechanism between global and local tokens, and enables the model scale to long inputs. However, the fully connected topology of its token graphs cannot capture the natural structure of the document.

(2) DocHopper (Sun et al., 2021) highlights the structural information that a passage contains consecutive and relevant information, and retrieves information by joint sentence and passage level. A structural information between passages is ignored

Dataset	*HotpotQA-Doc*		*Qasper*		*ConditionalQ*	
Settings	*Evidence*	*Answer*	*Extractive*	*Abstractive*	*Extractive*	*Conditional*
gpt-3.5-turbo		41.0	54.9	**27.8**		
ETC					17.3	41.8
DocHopper					26.7	46.4
FID					37.8	49.7
SDHD					42.0	**52.3**
D^3	**26.9**	**43.5**	42.9	23.7		
MedDiscourse (ours)	23.2	42.0	**56.4**	24.7	**44.2**	47.1

Table 2: Comparative performance of QA against long documents

(3) FID (Izacard and Grave, 2021) independently encodes different passages and concatenates the representations in the decoder only, which decreases calculation cost and improves performance for QA on long documents. However, the natural structure of documents and discourse information in each section are neglected.

(4) SDHG (Structure-Discourse Hierarchical Graph, Du et al. 2023) conducts bottom-up information propagation, firstly by building the sentence-level discourse graphs for each section and encoding the discourse relations by graph attention. Secondly, a section-level structure graph is built based on natural structures, and conducts interactions over the question and contexts. Finally, different levels of representations are integrated into jointly answer and condition decoding.

(5) D3 (Nair et al., 2023) utilizes discourse structure commonly found in documents

We show F1 accuracies in answering questions (Table 2). One can observe that the proposed system outperforms the other long-document QA in a Conditional Q-Extractive and Qasper-Extractive evaluation settings. For HotpotQA-Doc, the performance of D3 is systematically better. At the same time, SDHD shows a superior performance in the case of Conditional evaluation.

Identifying Hallucinations

The third problem for shaped charge learning is hallucination identification in a content generated by a DNN-based system. Once identified, we repair these hallucinations with factual content taken from various sources. We use the structure and content flow from the raw text, while replacing each hallucinating phrase with the one with factual information gathered from the web. This enables us to create original content produced by a DNN with factual truthfulness provided by the web or other reliable source of information.

For each sentence in the raw text, we perform a deterministic fact-checking process as an implementation of kNN learning. We iterate through each sentence and make modifications to the syntactic structure as little as possible. We first use syntactic criteria to determine if the sentence should be retained in the corrected content. We then proceed to fact-checking to form a family of queries from the sentence to obtain true candidate sentences. These true sentences are extracted from search result snippets and identified documents, and then matched against the given raw sentences to determine what is the most optimal substitution. Syntactic and semantic alignments are built and the entities and phrases to be substituted are identified.

The architecture chart Fig. 6 shows major decisions regarding whether the raw sentence can be modified; if yes, should it be an entity substitution or a phrase substitution? The structure of coreferences in the raw text must be taken into account when making substitutions for multiple sentences. The discourse structure of the resulting text should then be compared to that of the original text.

We measure the rate of errors per whole text (10 sentences on average) and per sentence for both raw LLM and hallucination-processed texts (Galitsky 2023). We observe that the hallucination rate is 15.91/6.60 = 2.4 times less frequent as a result. We also measure the rate of discourse distortion, i.e., the percent of cases where the hallucination correction ended up breaking the overall logic or organization of a text. It happens in about 1/5 of all texts.

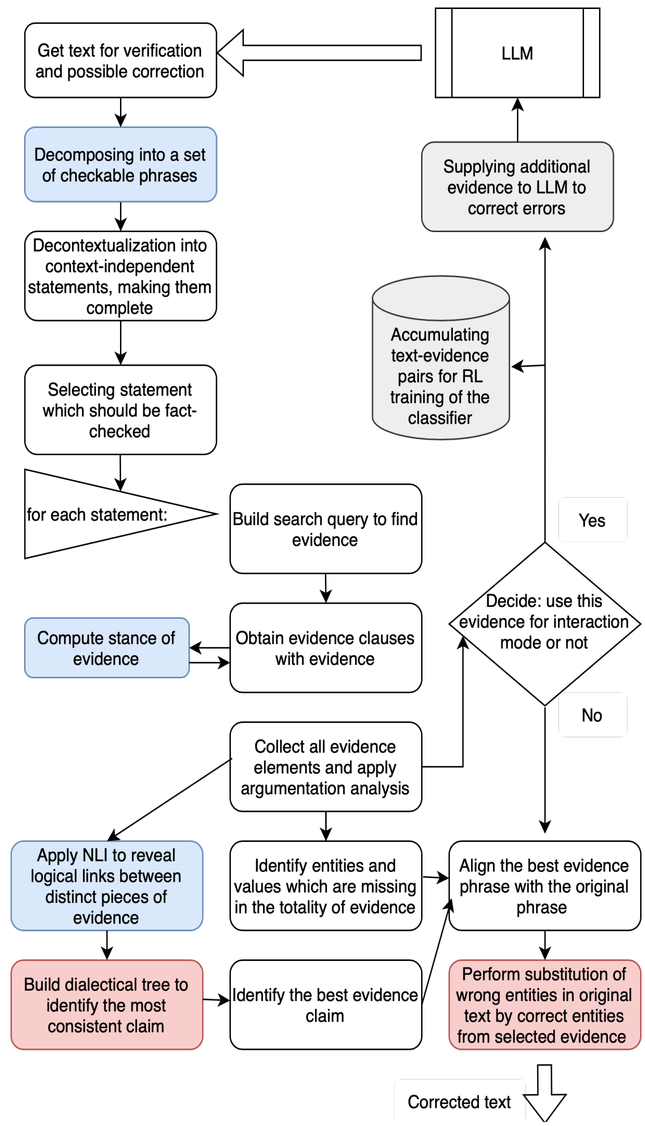

Figure 6: Hallucination correction architecture instance of shaped-charge learning

	Movielens			Recipe			Yelp		Sports	
	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10	HR@5	NDCG@10	HR@5	NDCG@10
MLP	0.291	0.244	0.363	0.032	0.068	0.058				
LLM-Rec	0.315	0.277	0.395	0.039	0.084	0.070				
FMLP-Rec							0.273	0.502	**0.380**	0.316
PALR		0.072*	0.044*							
Ours baseline (LLM finetuning)	0.307	0.210	0.320	0.029	0.065	0.043	0.28	0.460	0.355	0.261
Ours with abduction	**0.321**	**0.254**	0.390	**0.041**	0.062	**0.075**	**0.291**	**0.489**	0.371	**0.332**
Ours with meta prompts	0.304	0.231	**0.403**	0.0361	**0.072**	0.068	0.273	0.465	0.376	0.309

Table 3: Results of LLM personalization

Providing Personalized Recommendation

Today, LLMs are not good at personalization providing a recommendation. They advise physicians and financial advisors to ask professionals in respective fields for help, even having user information available. Answering questions of software professionals, LLMs need to deliver in-depth answers with codes or algorithms, whereas professionals in other fields would need definitions and main concepts. In this section we evaluate a shape-charge learning based approach to adjusting LLM answers to the needs of users, taking into account available information about them.

We apply logical abduction to adjust the prediction results for a given user, taking into account her personalization profile. Default, unpersonalized results are delivered by an LLM. Then the abduction component (instead of a kNN in a default shaped-charge architecture) performs reasoning about the needs of a particular user.

To do that, we need to generalize available information about a person like her health record, while maintaining the privacy of this person. We rely on meta-learning techniques to design an LLM prompt to produce a personalization prompt to obtain suitable relevant information. Such “meta-prompt” is produced by generalization operation applied to available documents for the user. These documents need to be de-identified so that they are sufficient for personalization on the one hand and will maintain user privacy on the other hand. The second neuro-symbolic technique to support personalization is abductive reasoning, acting in parallel to LLM fine-tuning.

We compare Multi-layered perceptron (MLP in Zhou et al. 2022), PALR (Chen et al. 2023), LLM-Rec (Lyu et al. 2023) and FMLP-Rec (Zhou et al. 2022) in the four datasets of sequential item-based recommendations: Movielens, Recipe, Yelp and Sports.

We show three types of performances for our system on popular datasets (Table 3):

(1) Our baseline with LLM finetuning.
(2) Baseline extended with abduction;
(3) Baseline extended with meta prompts.

One can observe that our baseline performance is below all the competitive systems. We were not able to match FMLP-Rec on Sports database in our approach but improved its performance in the Yelp domain. The best performance is achieved using abduction, and the second best – using meta prompts, in comparison with LLM-Rec, FMLP-Rec and PALR. Abduction approach is the winner in the Movielens dataset, outside of NDCG measure, and also in the Recipe dataset, outside of the Recall@10 measure. We observe that abduction and meta prompts gives a substantial boost to personalization performance.

These are promising directions to improve overall recommendation accuracy as LLMs are being further developed.

Conclusions

Significant endeavors in the AI community have resulted in the creation of neurosymbolic systems that integrate ML, DNN, and automated reasoning. These systems have found application in diverse fields, such as computational biology, fault diagnosis, simulator-based training and assessment, and software verification (Besold et al., 2017).

The fusion of DNN and explainable ML approaches, both employing gradient descent (Domingos, 2020), addresses the inherent lack of explainability in DNN predictions. This approach ensures a seamless transition from the former (DNN) to the latter (explainable ML in Goldberg et al. 2021) during a prediction session. Recognizing that deep networks essentially function as path kernel machines significantly enhances interpretability. The synaptic connection weights in an LLM can be directly interpreted as a superposition of training samples in a gradient space, where each sample can be mapped to the corresponding gradient of a model. By combining deep and nearest-neighbor learning under the single gradient descent umbrella, this approach harnesses the high-level performance of the former and the interpretability of the latter. Additionally, the incorporation of meta-learning control automates procedures related to feature engineering and training dataset manipulation.

We observed that shaped-charge learning as an instance of neurosymbolic architecture yields substantial improvement in a broad range of NLP systems including question-answering, discourse processing, hallucination detection, and personalization.

References

Ainslie J, Ontanon S, Alberti C, VCvicek V, Fisher Z, Pham P, Ravula A, Sanghai S, Wang Q, and Yang L. 2020. Etc: Encoding long and structured inputs in transformers. In EMNLP, pages 268–284.

Besold, T.R., Garcez, A.S., Bader, S., Bowman, H., Domingos, P.M., Hitzler, P., Kühnberger, K., Lamb, L., Lowd, D., Lima, P.M., Penning, L.D., Pinkas, G., Poon, H., & Zaverucha, G. 2017. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. ArXiv, abs/1711.03902.

Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., Pu, Y., Lei, Y., Chen, X., Wang, X., Lian, D., & Chen, E. 2023. When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities. ArXiv, abs/2307.16376.

Domingos, P. Every Model Learned by Gradient Descent Is Approximately a Kernel Machine. arXiv 2020, arXiv:1905.10985.

Evans, R. and Grefenstette, E. 2018. Learning Explanatory Rules from Noisy Data. IJCAI 2018, 5598–5602.

Galitsky B. 2020. Employing Abstract Meaning Representation to Lay the Last-Mile Toward Reading Comprehension. In: Artificial Intelligence for Customer Relationship Management. Springer, Cham.

Galitsky B. Applications of Shaped-Charge Learning. 2023. Entropy.; 25(11):1496. https://doi.org/10.3390/e25111496

Galitsky, B.; Ilvovsky, D.; and Goldberg, S. 2023. Shaped-Charge Learning Architecture for the Human–Machine Teams. Entropy 2023, 25, 924. https://doi.org/10.3390/e25060924

Goldberg, S.; Pinsky, E.; Galitsky, B. 2021. A bi-directional adversarial explainability for decision support. Hum. Intell. Syst. Integr. 2021, 3, 1–14

Honavar, V. 1995. Symbolic artificial intelligence and numeric artificial neural networks: towards a resolution of the dichotomy. Computational architectures integrating neural and symbolic processes: a perspective on the state of the art , 351–388.

Izacard G and Grave E. 2021. Leveraging passage retrieval with generative models for open domain question answering. In Proceedings of the 16th Conference of the European Chapter of the ACL, 874–880

Kahneman, D. 2011. Thinking, fast and slow. Penguin Books Limited.

Lyu H and Song Jiang and Hanqing Zeng and Qifan Wang and Si Zhang and Ren Chen and Chris Leung and Jiajie Tang and Yinglong Xia and Jiebo Luo. 2023. LLM-Rec: Personalized Recommendation via Prompting Large Language Models, arXiv2307.15780

Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, arXiv:1904.12584

Marra, G., Diligenti, M., Giannini, F., Gori, M., Maggini, M. 2020. Relational neural machines.

Muggleton, S. (Ed.) 1992. Inductive Logic Programming; Academic Press: Cambridge, MA, USA,

Nienhuys-Cheng, S.-H.; Wolf, R.d.; Siekmann, J.; Carbonell, J.G. 1997. Foundations of Inductive Logic Programming; Springer: Berlin/Heidelberg, Germany.

Otto H.R., Tuedio J.A. 1998. Perspectives on Mind. Synthese Library (SYLI), volume 194.

Rumelhart D.E.; Hinton, G.E.; Williams, R.J. 1986. Learning representations by back- propagating errors. Nature, 323, 533–536

Safjan K. 2023 Explaining AI - The Key Differences Between LIME and SHAP Methods. Krystian's Safjan Blog https://safjan.com/explaining-ai-the-key-differences-between-lime-and-shap-methods. July 21, 2023

Sun H; Cohen W; and Salakhutdinov R. 2022. ConditionalQA: A complex reading comprehension dataset with conditional answers. ACL, pp 3627–3637

Sun, R., Bookman, L.A. 1994 Computational architectures integrating neural and symbolic processes: A perspective on the state of the art. Kluwer Academic Press.

Wang, W.Y.; Mazaitis, K.; William; Cohen, W. 2014. Structure learning via parameter learning. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014 pp. 1199–1208

Yu D, Bo Yang,Dayou Liu,Hui Wang,Shirui Pan. 2023. A Survey on Neural-symbolic Learning Systems. arXiv:2111.08164v3

Zhou K, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In Proceedings WWW '22, 2388–2399