Reinforcement Learning-based Knowledge Graph Reasoning for Explainable Fact-checking

Gustav Nikopensius
Institute of Computer Science
University of Tartu, Estonia
gustav.nikopensius@ut.ee

Mohit Mayank
Outplay, India
mohit.m@outplayhq.com

Orchid Chetia Phukan
Department of CSE
IIIT-Delhi, India
orchidp@iiitd.ac.in

Rajesh Sharma
Institute of Computer Science
University of Tartu, Estonia
rajesh.sharma@ut.ee

Abstract—**Fact-checking is a crucial task as it ensures the prevention of misinformation. However, manual fact-checking cannot keep up with the rate at which false information is generated and disseminated online. Automated fact-checking by machines is significantly quicker than by humans. But for better trust and transparency of these automated systems, explainability in the fact-checking process is necessary. Fact-checking often entails contrasting a factual assertion with a body of knowledge for such explanations. An effective way of representing knowledge is the Knowledge Graph (KG). There have been sufficient works proposed related to fact-checking with the usage of KG but not much focus is given to the application of reinforcement learning (RL) in such cases. To mitigate this gap, we propose an RL-based KG reasoning approach for explainable fact-checking. Extensive experiments on FB15K-277 and NELL-995 datasets reveal that reasoning over a KG is an effective way of producing human-readable explanations in the form of paths and classifications for fact claims. The RL reasoning agent computes a path that either proves or disproves a factual claim, but does not provide a verdict itself. A verdict is reached by a voting mechanism that utilizes paths produced by the agent. These paths can be presented to human readers so that they themselves can decide whether or not the provided evidence is convincing or not. This work will encourage works in this direction for incorporating RL for explainable fact-checking as it increases trustworthiness by providing a human-in-the-loop approach.**

Keywords: Explainable AI, Fact-checking, Knowledge Graph, Reinforcement learning, Misinformation detection.

I. Introduction

Social media platforms are not only used by billions for communication but has become an alternative platform (compared to TV, print media) to receive news [1]. These platforms have also being misused for spreading misinformation [2], [3] and its various forms such as fake news [4], [5], rumors [6], [7], misleading news [8], etc. With the sheer pace of misinformation generated manual fact-checking can not keep up. So, automated fact-checking must be employed to tackle the menace of misinformation. One of the most prominent ways is through the usage of machine learning (ML) techniques [9], [10]. The majority of such techniques handle the context of the misinformation for its detection. For example, investigating what kind of users propagate the misinformation [11] or processing the content with natural language processing (NLP) techniques to derive clues from the way the misinformation is written, etc. [12].

It is important to mention that the rise in misinformation has bring forth the focus on fact-checking as it helps in controlling the spread of misinformation [13]–[15]. Providing explanations for fact-checking tools leads to trust among the people and offer better usability. Examples of the works related to explanations include formulating the task of explanation generation as an extractive-abstractive summarization task using transformer models [16], [17]. These particular works deal with specific domains such as political news [16] and public health [17]. In addition, to use these approaches, the datasets used in these studies have been annotated with gold-standard justifications by journalists. This approach is not scalable as annotation is a costly task. Alternatively, Knowledge graphs (KGs) have also been employed to provide a mechanism in which reliable third-party sources such as scientific articles and Wikipedia are used for fact-checking [18]–[20]. However, the use of reinforcement learning (RL) in such cases hasn't garnered much attention.

In this work, we propose an RL-based KG reasoning approach for explainable fact-checking that employs an RL agent to produce explanations in the form of relevant knowledge from the KG in the form of a path. The paths are arrived at by a technique from the field of KG reasoning known as multi-hop reasoning where an RL agent hops over facts in an attempt to connect two nodes that correspond to real-world entities. The path describes the relation between those two entities and this path is used to classify a statement as true or false and also provides a human-readable explanation of the classification result.

To summarize, the main highlights of this work are as follows:

1) To the best of our knowledge, this is the first work that has used a multi-hop reasoning agent for explainable fact-checking. We use two KGs (FB15K-277 and NELL-995) that are commonly used in multi-hop reasoning tasks to evaluate the performance of the proposed model. The KGs represent real-world relationships, sets of true facts are gathered from these KGs, and corresponding subsets of false facts are generated to simulate false facts.
2) The results show that a multi-hop reasoning model can successfully check facts and provide highly human-readable explanations for the classifications. The agent

is also able to provide explanations and a basis for classification which is not directly tied to any specific fact-checking task.

The rest of the paper is organized as follows. Section II discusses the related works. Next, in Section III we describe the methodology of the proposed approach, followed by the experiments and its results in Section IV. We conclude in Section V with some future directions.

II. Related Work

This section describes previous works related to multi-hop reasoning (Section II-A) and fact-checking with KG (Section II-B).

A. Multi-hop reasoning

KG reasoning infers new knowledge from existing knowledge in the KG, and detects errors in the KG. Large scale KGs are inherently incomplete. The main effort in KG reasoning research is to "fill in" the missing knowledge. This is done by either KG traversal known as multi-hop reasoning [21]–[23] or by mining logic rules based on KG semantics and using those rules for reasoning [24]–[26]. There are two main directions in multi-hop reasoning: entity prediction and relation prediction. The former looks for a tail entity for a head entity and relation pair and the latter looks for a missing relation between two entities. The task of looking for a missing tail entity is more similar to our work of finding a path that leads from a given head entity to the desired (true) tail entity. The major difference is that the proposed approach receives the claimed tail entity as an input as well.

Reasoning in the continuous vector space of KG embeddings was first proposed by Xiong et al. [27]. Compared to Das et al. [28] which rely on Path Ranking Algorithm (PRA) to produce the initial paths, Xiong et al. [27] utilizes a policy-based RL algorithm REINFORCE to learn paths in the KG. Das et al. [21] builds on Xiong et al. [27] by directing the task from relation prediction to link prediction, a more complicated task, and also building a model that can handle multiple reasoning entities. We use both the proposed target entity and relation as input, thus differing from both Das et al. [21] as well as Xiong et al. [27].

B. Fact-checking with Knowledge Graph

Ciampaglia et al. [29] framed the problem of fact-checking as finding a connection between the subject and object of a fact statement: assertion is true if there is an edge or short path between the entities in a KG, and false otherwise. Their proposed method makes use of a truth valuation function that accounts for the length of the path between the two entities as well as the generality of intermittent entities, where generality means *how many statements it participates in.* Shiralkar et al. [30] builds upon the notion of generality of intermittent entities but considers multiple paths between the subject and object entities as a knowledge stream. These approaches rely on pre-extracted subgraphs. Inversely, the proposed approach saves resources by integrating relevant subgraph extraction as a part of the explainable classification process.

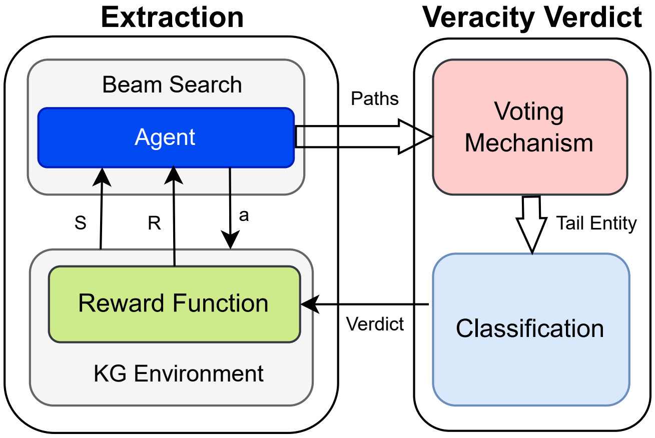

Fig. 1: Proposed Framework; "S" stands for State which contains the input claim and the path that is built starting from the head entity, "R" stands for the Reward which is used to update the agent's policy function at the end of the episode, "a" stands for the action which the Agent sends to the environment

Previous research is also involved with mining the KG for patterns to be used in classifying a fact statement. Shi et al. [31] mined the KG for patterns associated with specific relations. Lin et al. [32] used a supervised pattern discovery algorithm to find patterns relevant to training data. Lin et al. [33] builds upon this work by also taking into account the ontological closeness of entities. Unlike these works, this work wholly operates in the continuous vector space and is not formally concerned with patterns, although patterns do arise naturally from the RL aspect.

Explainable fact-checking with Knowledge Graphs: Ahmadi et al. [18] used KGs to assess claims and accompany these with human-readable explanations. This is achieved through a combination of logical rule discovery and probabilistic answer set programming (ASP). The KG is mined for rules in the form of triples associated with specific relations and these relations' negations. For fact assessment the heads of these rules are replaced by claim values and the KG is scavenged for corresponding triples. Explanations and an assessment are derived from these triples through ASP. Gad-Elrab et al. [19] used rules to extract explanations from KGs and text resources. Schatz et al. [20] mined for patterns for fact-discovery in the biomedical domain to produce human-readable explanations to professionals for drug or treatment discovery. The main difference between these previous approaches and the proposed approach is that it does not rely on mined patterns but a learned policy and KG embeddings.

III. Methodology

This section provides an overview of the proposed framework for explainable fact-checking (Figure 1). We first explain the path extraction process (Section III-A) succeeded by the veracity verdict (Section III-B), and the environment in Section III-C.

A. Evidential path extraction

The path extraction process follows multi-hop reasoning practices in which a policy-based agent sequentially extends an inference path. In contrast to traditional multi-hop reasoning tasks where one part of the claim is hidden from the agent, the proposed framework presents the agent with the entire claim and, in addition, the inference path, as it is being formed, as observations. We boost the quantity and versatility of extracted evidential paths by combining the policy-based agent with the beam search algorithm.

Policy-based agent: The policy-based agent receives a claim as input and, in return, extracts a path from the KG. The path extraction process follows the pattern of a Markov decision process (MDP). The agent considers an observation, comprising the input claim and current path, to decide upon an action with which to extend the current path, after which the path is updated, and the agent receives an updated observation. The process is repeated until the expiration of a predetermined number of time steps. The process results in a path that begins with the claim's head entity and bears, after sufficient training, a semantic similarity to the claim. And thus, the resulting path can be used to judge upon the veracity of the input claim. As semantically similar, the tail entities of the claim and the evidential paths should match for a true claim and differ for a false claim. The policy function maps the observation to a probability distribution over the action space which comprises all of the unique relations within the KG. The distribution is then filtered to only include relations that connect the current tail entity of the path. This is done to only include relations that can be used to extend the path. A *self-loop* relation is included within the action space in case the agent decides to remain at the current tail entity. The policy function is parametrized by a neural network to maximize the expected reward for the path-based classification of the claim.

Beam search: The beam search algorithm allows us to increase the number of paths which are extracted during an episode. It is customary to guide the relation choice in KG reasoning tasks, but to leave the entity choice to random chance. In this work, we propose a heuristic function that takes into account the fitness of the entities which the relations lead to by leveraging the KG scoring function [34]. By doing this, the search algorithm takes into account how fitting the resulting entity is as a correct tail entity for the evidential path by scoring it as the tail entity in the input triple. As the result of the scoring function is less important in the early steps of path extension, this function is the base of an exponential function, where the step number is the exponent. This function is added to the probability calculated for a given action by the policy function and the sum is used to compare viable path extension options to each other in the beam search process. The final equation stands (Equation 1):

$$\pi(a|s,\theta) + Re(< r, e_s, \bar{e}_c >)^{step} \tag{1}$$

where, $\pi(a|s,\theta)$ is the output probability of the policy function for the action a, given state s and policy function parameters θ, and $Re(< r, e_s, \bar{e}_c >)$ is the scoring function and $step$ is the step of the MDP episode. Here, r, e_s, and $\bar{e}_c$ are vector representations of relation, head entity, and tail entity. The tuple signifies the dot product of the three vectors. Re means the real part of the dot product.

B. Veracity verdict

The veracity verdict is arrived upon by grouping the extracted paths and counting all unique path tail entities. The straightforward approach would be to choose the most popular tail entity as the choice tail entity to be compared to the claim tail, but since the result can be a tie, a heuristic weighting is added to the votes. The beam search heuristic for the final step of each path is used to weight the vote of each entity. This practically excludes the possibility of a tie between the contending tail entities and adds leverage to the entities which the agent is more confident in and which received a higher score from the KG triple scoring function which is used to evaluate whether a given triple belongs to the KG or not and is used to fit vector embeddings to entities and relations. The verdict is then evaluated against the ground truth and the result is used to update the agent which extracted the paths.

C. Environment

The environment is modeled as a Markov decision process (MDP) which controls the interaction between the RL agent (the Agent) and the knowledge graph. A 4-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} >$ represents the MDP. Each element is described below.

State ($\mathcal{S}$): The state consists of entities and relations. Two representations of the state are operated with in parallel: 1) a human-readable collection of real-world elements: entities and relations, and 2) a corresponding collection of embeddings - vector representations that capture the semantic information of the state elements. The state contains the fact claim, traversed path, and current location of the agent which is itself an entity. The initial state consists of an input triple which is the fact claim. Thus the initial state can be thought of constituting as, < *claimed Target*, *claimed Relation*, *beginningOfPath*, ...>

The embedded state is padded with zero vectors since the state is used as input to the policy network which requires a fixed-size input. These zero vectors are incrementally replaced by relation-entity embedding pairs during the agent's path traversal process. The final entity in the set of these pairs is of particular importance: this is the entity that is used to give a veracity verdict for the fact claim. So in essence, once the path traversal is completed then the veracity verdict stands as:

$$claimedTarget == finalEntity$$

Action ($\mathcal{A}$): Not only is the input size of the policy network fixed but so is the output size - it is always the same size as the action space. The policy network takes the state as input and computes probabilities for each action in the action space. This can be thought of as the agent making an estimate:

which actions are most likely to take me from the current entity to the desired final entity? During training time, one of the top three most probable actions are chosen for each transition. The choice is random with a distribution that is calculated from the output of the policy network to fit the new sample size. Random choice is introduced in order to reduce the likelihood of the agent getting stuck looking for sub-optimal paths brought forth by the local minima problem.

Transition ($\mathcal{T}$): Choosing a new action entails a transition of state. The transition itself is a function $T : SXA - > S$, which means that performing an action at a given state produces a new state. This new state is the input triple, a traversed path plus the choice action and a new current entity. The new current entity is a node in the KG that is connected to the previous current entity by the choice action. It is not uncommon for an entity to have multiple neighbors that are all connected to it via the same relation. Most previous literature concerning multi-hop reasoning picks a neighboring entity at random [21], [27], the focus being on the performance of the RL agent, unsuited for the task of choosing a candidate from a variable size set of entities.

Reward ($\mathcal{R}$): The agent is given a reward once the path traversal is complete. The reward is 1 if the final entity in the found path is equal to the true target. Otherwise, the reward is 0.

$$reward = \begin{cases} 1, & \text{if final entity = true target} \\ 0, & \text{otherwise} \end{cases}$$

IV. Experiments

Here, we first describe the datasets (Section IV-A) followed by details regarding implementation and hyperparameters (Section IV-B), results of our experimentations in Section IV-C, explainability of the proposed approach (Section IV-D), and lastly a discussion of the results of our experiments (Section IV-E).

A. Dataset

The experiments are conducted on two widely used benchmark datasets: NELL-995 [27] and FB15K-237 [35]. Table I describes the specifics pretraining to the two datasets. NELL-995 is generated from the 995th iteration of the NELL system [36] by selecting the triples with the top 200 most frequent relations. FB15K-237's facts are created from FB15K [37] with redundant relations removed. Both datasets contain inverse triples to aid with path finding, allowing the reasoning agent to step backward in the graph. We chose three reasoning tasks per dataset: *origin*, *tvLanguage*, *nationality* for FB15K-237, and *athletePlaysInLeague*, *athletePlaysForTeam*, *worksFor* for NELL-995. To evaluate the agent's performance, each reasoning task contains approximately ten negative examples for each positive sample, and the negative samples were generated by replacing a true claim's tail with a faked entity.

TABLE I: Statistics of the datasets under consideration; #Ent, #Rel, and #Fact denotes the number of unique entities, relations, and triples respectively

Dataset	#Ent	#Rel	#Fact
FB15k-237	14,505	237	272,115
NELL-995	75,492	200	154,213

B. Implementation Details and Hyperparameters

Knowledge Graph Embeddings: We use *ComplEx* [38] embedding model from *Ampligraph* [39] library for KG embeddings. The model's hyperparameters are chosen to balance model complexity and training time. We set the batch size as 50 and train the model for 1000 epochs. The embedding dimension is set to 20. We use Adam optimizer with a learning rate of 1e-4 and negative log-likelihood loss function. To prevent overfitting, we use an L3 regularization term with a regularization strength of 1e-5. We also set the random seed to 0 to ensure the reproducibility of the results.

Policy Network: We use REINFORCE algorithm to train the policy network and *PyTorch* library for the implementation. It comprises of ReLU as an activation function in the intermediate layers and Softmax in the output layer to generate a probability distribution over the actions given the current state. The network has hidden state size of 128 neurons and was initialized using the Xavier initialization method. We train the network for 100,000 episodes for each reasoning task, each consisting of 3 steps. We use *Adam* optimizer with a learning rate of 0.001. To encourage exploration, we randomly chose one of the top three most likely actions as the action to take in each state during training.

During the evaluation, we use beam search to obtain multiple likely actions according to the policy network's probability distribution for each state. This allows us to explore multiple possible paths through the action space and obtain a more robust estimate of the policy's performance. Three beam numbers, 3, 5, and 10 are considered. We assess the performance of our model using an 80:20 train-test split. We use hits@k metric, which is commonly used in knowledge graph reasoning tasks. The hits@k metric measures the proportion of test samples where the correct answer is among the top k-ranked answers returned by the model. In addition, we use accuracy as a metric for the voting mechanism. This was calculated as the proportion of test samples where the weighted majority vote was the correct answer. We acknowledge that the varying degrees of information in the knowledge graphs are dependent on alternative evidential paths. However, setting thresholds to consider the number of alternative paths for each reasoning task is complex and requires further research. As such, we did not take this into account in our evaluation.

C. Experimental Results

Basis for interpreting the results: The two KGs used in experimentation are different, particularly in terms of their density features, with FB15K-237 containing 18.8 triples per entity and NELL-995 containing two triples per entity. This difference may affect the model's accuracy. The subsets for both KGs consist of 3 single relation tasks, with each KG also having a combined subset containing all three relations. This diversity in task assignment may complicate the agent's task, but it seems to actually assist the agent by giving meaning to the input triple's relation. In single relation tasks, the input triple's relation never changes and therefore does not inform the agent's actions.

Results: Tables II and III show the results of our proposed approach on Hits@k and Voting Accuracy for FB15K-237. Similarly, Tables IV and V show the results of our proposed approach on Hits@k and Voting Accuracy for NELL-995. The results show that the method is able to achieve high accuracy in finding a path to the true target entity with a high enough beam number, as indicated by the results of the hits@k metric. A higher beam number means more unique paths are considered, increasing the probability of reaching the target entity. Table II shows that considering more paths increases the proportion of input triples for which at least one path reached the target entity. But that's not the case always, considering more paths can even decrease accuracy because it requires the agent to discard more paths as is the case for *origin* in Table III, and *athleteplaysinleague* and *worksfor* in Table V. However, the accuracy drops slightly after the voting process, as the correct path must be selected from the paths found. In particular, the "origin" dataset in FB15K-237 (Table III) has shown to be anomalous, achieving lower accuracy than expected despite being a larger dataset. The voting task has proven to positively impact accuracy, especially for the combined dataset in FB15K-237, indicating the model's scalability. Nonetheless, the results demonstrate that voting on an answer reduces accuracy, as the correct path and target entity may be correctly identified but still be undermined by a higher majority of an entity in the vicinity of the source entity. Both FB15K-237 and NELL-995 follow similar trends in terms of results regarding beam numbers and an accuracy trade-off in voting accuracy.

TABLE II: Hits@k for FB15K-237; acc stands for accuracy

	dataset size	acc., b=3	acc., b=5	acc., b=10
origin	852	0.464	0.53	0.661
tvLanguage	614	0.52	0.659	0.894
nationality	7134	0.866	0.907	0.921
combined	8600	0.807	0.886	0.943

TABLE III: Voting accuracy for FB15K-237; acc stands for accuracy

FB15K-237	dataset size	acc., b=3	acc., b=5	acc., b=10
origin	852	0.357	0.369	0.292
tvLanguage	614	0.463	0.472	0.642
nationality	7134	0.727	0.75	0.765
combined	8600	0.252	0.602	0.727

TABLE IV: Hits@k for NELL-995; acc stands for accuracy

NELL-995	dataset size	acc, b=3	acc, b=5	acc, b=10
athleteplaysinleague	12660	0.919	0.971	0.994
athleteplayssport	6781	0.903	0.853	0.868
worksfor	9669	0.748	0.78	0.825
combined	29110	0.485	0.575	0.611

TABLE V: Voting accuracy for NELL-995; acc stands for accuracy

NELL-995	dataset size	acc, b=3	acc, b=5	acc, b=10
athleteplaysinleague	12660	0.858	0.938	0.875
athleteplayssport	6781	0.871	0.828	0.83
worksfor	9669	0.604	0.59	0.509
combined	29110	0.45	0.467	0.464

D. Explainability

For the purpose of providing explanations, we present examples of single-relation tasks. Each example is evaluated based on the proposed approach's performance in arriving at the correct conclusion, the informativeness of the path, and the ground truth interpretation. We provide six examples in total.

Example 1: KG contains outdated information

Claim: Dick Cheney works for retailstore Halliburton

Path: Dick Cheney – leads organization → retailstore Halliburton

The agent has found a correlation between a person leading an organization and working for the organization. This is an interesting successful case, since Dick Cheney stopped leading Halliburton the day George W. Bush took him on as vice president candidate for the 2000 US presidential elections. This means the presented ground truth is actually false due to outdated information in the KG.

Example 2: Inconclusive path

Claim: Coach John Gruden works for sportsteam Buccaneers.

Path: Coach John Gruden ←organization hired person– sportsteam Buccaneers

The agent attempts to find a solution by considering the relation "organization hired person". In fact Tampa Bay Buccaneers had hired John Gruden in 2002 and Gruden worked for the team until 2008. Thus, the claim that John Gruden works for the Buccaneers was true from 2002-2008, but is not true anymore, meaning the agent's reasoning is good, but the conclusion arrived at is false, indicating the need for further investigation. The agent has learned a correlation between the relations "worksfor" and "organizationhiredperson inv" meaning if an organization has hired a person, then the person works for that organization.

Example 3: Informative path with complex reasoning

Claim: Coach Brendan Shanahan plays Hockey.

Path: Brendan Shanahan –plays for team→ the Devils –team plays sport→ Hockey

This is a good example of the agent finding a path that does not just contain a (near) synonym for the claimed relation (like *works for* and *leads organization*), but finding a more complicated path. The agent finds that Brendan Shanahan plays for the Devils hockey team, and then finds out that the Devils play hockey.

Example 4: Agent stays on the fence

Claim: Journalist Amir Taheri works for Los Angeles County.
Path: Amir Taheri → Amir Taheri
The Agent is unable to move beyond the source entity, resulting in an inconclusive path. This may be possible because of three reasons:

1) The length of the path is fixed to three relations, no more, no less.
2) The agent must have the possibility to decide upon a target entity and "stick with it"
3) The agent can decide to stay put at an entity for all three transitions.

Since most resultative paths only take one or two hops from entity to entity, the action of staying put is very common. Thus it's always one of the top actions to choose for any state. With beam search also in place, it is very common that one path will result in the agent staying put for the entirety of the path.

Example 5: Two true paths with only one interpreted as correct

Claim: Ben Folds originates from Nashville.
Path 1: Ben Folds –place of birth→Winston-Salem
Path 2: Ben Fold –location→ Nashville

These paths state that Ben Folds was born in Winston-Salem and that he is located in Nashville. For this specific case, the ground truth is that Ben Folds originates from Winston-Salem, so the first path leads to the true target and the second path reinforces the "false claim". The majority of explainable paths for the "origin" subset consist of one of two (near) synonymous relations: "place of birth" and "location". It is also common for one of the path relations to lead to a correct target entity and the other to lead to an incorrect one, the meaning of "origin" staying quite ambiguous. The subset is concerned with musical artists, so sometimes "origin" will mean the artists place of birth, other times it will mean the city in which they became established as a musician which will often coincide with the "location" relation.

Example 6: Two true explainable paths with only one being considered as correct

Claim: Kylie Minogue originates from Australia.
Path 1: Kylie Minogue –place of birth→ Melbourne
Path 2: Kylie Minogue –place of birth→ Melbourne ←contains– Australia

The claim that Kylie Minogue originates from Australia is true in real life, but in regard to the ground truth, it is considered false. Kylie Minogue originates from Australia which is true, but the ground truth is that Kylie Minogue originates from Melbourne, which is a city in Australia. Due to the setup of the veracity verdict mechanism, a path that concludes that Kylie Minogue originates from Australia is be considered false even though the path contains Melbourne and is very much explainable.

E. Discussion

The results of this study demonstrate that selecting an entity to be evaluated for veracity has a trade-off with accuracy. While the search algorithm may have identified the correct entity, it may not have been selected by the voting mechanism. This trade-off is not linear, and tasks with lower accuracy in identifying a correct path are more severely impacted by the results of voting. Analysis of common patterns in the paths generated by the agent highlighted several issues that should be considered in future research on this problem.

Firstly, it is crucial to use an up-to-date KG to ensure the accuracy of fact classifications. Outdated information in a KG can produce false results. Secondly, the verdict mechanism would benefit from identifying and incorporating inconclusive paths. The agent's motivation to move away from the source entity should also be increased. Additionally, paths that do not end with the target entity but support the ground truth should also be utilized to arrive at a correct conclusion. Lastly, the classification process could be improved by learning to deal with ambiguous relationships such as "origin". These findings suggest several avenues for future research to improve the accuracy and efficiency of veracity verdicts.

V. Conclusion

This work proposes a combination of reinforcement learning and KG to provide explainable fact classifications. The problem was formatted as a multi-hop reasoning problem enhanced by a beam search-based voting mechanism, which uses the paths produced by a graph hopping reinforcement learning agent to vote on the veracity of a claimed fact statement. The results show that this sort of approach can be used for fact classification with varying accuracy dependent on the nature of the input claims, and it produces highly human-readable explanations for the classifications. Future directions will make use of more massive KGs and extract facts-to-check from fake news datasets to provide a highly usable framework for real-world misinformation detection tasks.

VI. Acknowledgment

This work has received funding from the EU H2020 program under the SoBigData++ project (grant agreement No. 871042), by the CHIST-ERA grant No. CHIST-ERA-19-XAI-010, (ETAg grant No. SLTAT21096), and partially funded by HAMISON project.

References

[1] P. J. Boczkowski, E. Mitchelstein, and M. Matassi, "Incidental news: How young people consume news on social media," in *HICSS*, 2017.

[2] M. Flintham, C. Karner, K. Bachour, H. Creswick, N. Gupta, and S. Moran, "Falling for fake news: Investigating the consumption of news via social media," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–10. [Online]. Available: https://doi.org/10.1145/3173574.3173950

[3] R. Jagtap, A. Kumar, R. Goel, S. Sharma, R. Sharma, and C. P. George, "Misinformation detection on youtube using video captions," *arXiv preprint arXiv:2107.00941*, 2021.

[4] M. Mayank, S. Sharma, and R. Sharma, “Deap-faked: Knowledge graph based approach for fake news detection,” in *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2022, pp. 47–51.

[5] M. Dhawan, S. Sharma, A. Kadam, R. Sharma, and P. Kumaraguru, “Game-on: Graph attention network based multimodal fusion for fake news detection,” *arXiv preprint arXiv:2202.12478*, 2022.

[6] S. Butt, S. Sharma, R. Sharma, G. Sidorov, and A. Gelbukh, “What goes on inside rumour and non-rumour tweets and their reactions: A psycholinguistic analyses,” *Computers in Human Behavior*, vol. 135, p. 107345, 2022.

[7] S. Sharma and R. Sharma, “Identifying possible rumor spreaders on twitter: A weak supervised learning approach,” in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[8] S. Sharma, R. Sharma, and A. Datta, “(mis) leading the covid-19 vaccination discourse on twitter: An exploratory study of infodemic around the pandemic,” *IEEE Transactions on Computational Social Systems*, 2022.

[9] M. R. Islam, S. Liu, X. Wang, and G. Xu, “Deep learning for misinformation detection on online social networks: a survey and new perspectives,” *Social Network Analysis and Mining*, vol. 10, pp. 1–20, 2020.

[10] S. Sharma, A. Datta, V. Shankaran, and R. Sharma, “Misinformation concierge: A proof-of-concept with curated twitter dataset on covid-19 vaccination,” in *CIKM*, 2023.

[11] A. Ghenai and Y. Mejova, “Fake cures: user-centric modeling of health misinformation in social media,” *Proceedings of the ACM on human-computer interaction*, vol. 2, no. CSCW, pp. 1–20, 2018.

[12] Q. Su, M. Wan, X. Liu, C.-R. Huang *et al.*, “Motivations, methods and metrics of misinformation detection: an nlp perspective,” *Natural Language Processing Research*, vol. 1, no. 1-2, pp. 1–13, 2020.

[13] A. Martín, J. Huertas-Tato, Á. Huertas-García, G. Villar-Rodríguez, and D. Camacho, “Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference,” *Knowledge-based systems*, vol. 251, p. 109265, 2022.

[14] K. Tymoshenko and A. Moschitti, “Strong and light baseline models for fact-checking joint inference,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 4824–4830.

[15] S. Sharma, E. Agrawal, R. Sharma, and A. Datta, “Facov: Covid-19 viral news and rumors fact-check articles dataset,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, 2022, pp. 1312–1321.

[16] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, “Generating fact checking explanations,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7352–7364.

[17] N. Kotonya and F. Toni, “Explainable automated fact-checking for public health claims,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 2020, pp. 7740–7754.

[18] N. Ahmadi, J. Lee, P. Papotti, and M. Saeed, “Explainable fact checking with probabilistic answer set programming,” in *Proceedings of the 2019 Truth and Trust Online Conference (TTO 2019), London, UK, October 4-5, 2019*, M. Liakata and A. Vlachos, Eds., 2019. [Online]. Available: https://truthandtrustonline.com/wp-content/uploads/2019/09/paper_15.pdf

[19] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, “Exfakt: A framework for explaining facts over knowledge graphs and text,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*. ACM, 2019, pp. 87–95.

[20] K. Schatz, C. C. Melo-Filho, A. Tropsha, and R. Chirkova, “Explaining drug-discovery hypotheses using knowledge-graph patterns,” in *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021*. IEEE, 2021, pp. 3709–3716.

[21] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, and A. McCallum, “Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning,” *arXiv*, no. 1711.05851, Nov. 2017.

[22] X. Lv, X. Han, L. Hou, J. Li, Z. Liu, W. Zhang, Y. Zhang, H. Kong, and S. Wu, “Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph,” *CoRR*, vol. abs/2010.01899, 2020. [Online]. Available: https://arxiv.org/abs/2010.01899

[23] G. Wan and B. Du, “Gaussianpath: A bayesian multi-hop reasoning framework for knowledge graph reasoning,” in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021*. AAAI Press, 2021, pp. 4393–4401.

[24] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen, “Iteratively learning embeddings and rules for knowledge graph reasoning,” in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 2019, pp. 2366–2377.

[25] M. Qu, J. Chen, L. A. C. Xhonneux, Y. Bengio, and J. Tang, “Rnnlogic: Learning logic rules for reasoning on knowledge graphs,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=tGZu6DlbreV

[26] P. Minervini, S. Riedel, P. Stenetorp, E. Grefenstette, and T. Rocktäschel, “Learning reasoning strategies in end-to-end differentiable proving,” in *Neuro-Symbolic Artificial Intelligence: The State of the Art*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2021, vol. 342, pp. 280–293.

[27] W. Xiong, T. Hoang, and W. Y. Wang, “Deeppath: A reinforcement learning method for knowledge graph reasoning,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 564–573.

[28] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, “Chains of reasoning over entities, relations, and text using recurrent neural networks,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, 2017, pp. 132–141.

[29] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, “Computational fact checking from knowledge networks,” *CoRR*, vol. abs/1501.03471, 2015. [Online]. Available: http://arxiv.org/abs/1501.03471

[30] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, “Finding streams in knowledge graphs to support fact checking,” in *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*, V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, Eds. IEEE Computer Society, 2017, pp. 859–864.

[31] B. Shi and T. Weninger, “Discriminative predicate path mining for fact checking in knowledge graphs,” *Knowl. Based Syst.*, vol. 104, pp. 123–133, 2016.

[32] P. Lin, Q. Song, Y. Wu, and J. Pi, “Discovering patterns for fact checking in knowledge graphs,” *ACM J. Data Inf. Qual.*, vol. 11, no. 3, pp. 13:1–13:27, 2019.

[33] P. Lin, Q. Song, and Y. Wu, “Fact checking in knowledge graphs with ontological subgraph patterns,” *Data Sci. Eng.*, vol. 3, no. 4, pp. 341–358, 2018.

[34] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *International conference on machine learning*, 2016, pp. 2071–2080.

[35] K. Toutanova and D. Chen, “Observed versus latent features for knowledge base and text inference,” in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 57–66. [Online]. Available: https://aclanthology.org/W15-4007

[36] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI’10. AAAI Press, 2010, p. 1306–1313.

[37] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013, pp. 2787–2795.

[38] T. Trouillon, J. Welbl, S. Riedel, Éric Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” Jun. 2016.

[39] L. Costabello, A. Bernardi, A. Janik, S. Pai, C. L. Van, R. McGrath, N. McCarthy, and P. Tabacof, “AmpliGraph: a Library for Representation Learning on Knowledge Graphs,” Mar. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2595043