Unsupervised Fact Checking by Counter-Weighted Positive and Negative Evidential Paths in A Knowledge Graph

Jiseong Kim
Semantic Web Research Center
School of Computing
KAIST
jiseong@kaist.ac.kr

Key-Sun Choi
Semantic Web Research Center
School of Computing
KAIST
kschoi@kaist.ac.kr

Abstract

Misinformation spreads across media, community, and knowledge graphs in the Web by not only human agents but also information extraction algorithms that extract factual statements from unstructured textual data to populate the existing knowledge graphs. Traditional fact checking by experts or crowds is increasingly difficult to keep pace with the volume of newly created misinformation in the Web. Therefore, it is important and necessary to enhance the computational ability to determine whether a given factual statement is truthful or not. We view this problem as a truth scoring task in a knowledge graph. We present a novel rule-based approach that finds positive and negative evidential paths in a knowledge graph for a given factual statement, and calculates a truth score for the given statement by unsupervised ensemble of the found positive and negative evidential paths. For example, we can determine the factual statement "*United States is the birth place of Barack Obama*" as truthful if there is the positive evidential path (`Barack Obama`, `birthPlace`, `Hawaii`) ∧ (`Hawaii`, `country`, `United States`) in a knowledge graph. For another example, we can determine the factual statement "*Canada is the nationality of Barack Obama*" as untruthful if there is the negative evidential path (`Barack Obama`, `nationality`, `United States`) ∧ (`United States`, ≠, `Canada`) in a knowledge graph. For evaluating on a real-world situation, we constructed an evaluation dataset by labeling truth or untruth label on factual statements that were extracted from Wikipedia texts by using the state-of-the-art BERT-based information extraction system. Our evaluation results show that our approach outperforms the state-of-the-art unsupervised approaches significantly by up to 0.12 AUC-ROC and even outperforms the supervised approach by up to 0.05 AUC-ROC not only in our dataset but also in the two different standard datasets.

1 Introduction

Misinformation in the Web creates a situation in which false statements compete for attention to true statement necessary for users and applications. Misinformation in media and community makes difficult for users to search the information they need and misinformation in knowledge graphs makes difficult for applications to get the outputs they expect. This problem is common and getting worse in modern digital society. Although a lot of information in the Web is a good resource, there is certainly no guarantee that a given factual statement is truth or not. In order not to be fooled by false statements, it is necessary to separate truth from untruth by assessing truthfulness of factual statements.

We represent a factual statement as a triple (`subject`, `predicate`, `object`) where `subject` and `object` are entities that have a relationship between them as indicated by `predicate`. For example, "*Leonardo da Vinci is known for Mona Lisa*" can be represented as (`Leonardo da Vinci`, `knownFor`, `Mona Lisa`). A set of such triples is called a knowledge graph where nodes represent the entities and directed edges represent the predicates. Different predicates can be represented by edge

Proceedings of the 28th International Conference on Computational Linguistics, pages 1677–1686
Barcelona, Spain (Online), December 8-13, 2020

types. If a knowledge graph was complete to know all the facts, the fact checking would be as easy as checking whether a given factual statement is contained in a knowledge graph or not. In reality, a knowledge graph has limited and sparse information.

Some information extraction algorithms (Nam et al., 2020; Min et al., 2017) try to populate incomplete knowledge graphs by finding missing facts from unstructured textual data in the Web. However, as the information extraction task is challenging and an accuracy of such algorithms is not yet complete, they often produce incorrect outputs which result in corrupting a knowledge graph with false statements. Our dataset presented in this paper indicates that 83.51% of the factual statements extracted from Wikipedia texts by the state-of-the-art BERT-based information extractor are actually false statements. Our goal is to determine truthfulness of factual statements which can serve as a key clue to separate true statements from false ones. We view this problem as a truth scoring task in a knowledge graph which is to assign a truth score ranging from 0.0 (untruthful) to 1.0 (truthful) to a given factual statement based on supporting evidential paths that can be found in a knowledge graph.

In this paper, we present a novel rule-based unsupervised approach that uses positive and negative rules to find positive and negative evidential paths to calculate a truth score for a given factual statement. Our contributions are as follows: (1) An unsupervised fact checking approach that outperforms the state-of-the-art unsupervised approaches significantly by up to 0.12 AUC-ROC and even outperforms the supervised approach by up to 0.05 AUC-ROC in the three different datasets. (2) A novel counter-weight measure for a rule, which considers not only correct examples but also counter examples for calculating a weight, and is more effective for a truth scoring task by up to 0.2 AUC-ROC than the existing weight measures. We believe that this is the key factor for outperforming the existing supervised approach. (3) A novel negative sampling method based on Distant Local Closed World Assumption (D-LCWA), which is more effective for a truth scoring task by up to 0.05 AUC-ROC than the existing negative sampling methods. (4) A novel evaluation dataset for a fact checking problem, which comprises the factual statements missing in a knowledge graph. It makes our dataset more suitable than the existing datasets for evaluating the ability to validate additional facts missing in a knowledge graph.

2 Related Work

Approaches for truth scoring can be broadly classified into two types: (1) approaches that use unstructured textual data to find supporting evidential sentences for a given statement (Gerber et al., 2015; Syed et al., 2018; Thorne and Vlachos, 2018), and (2) approaches that use a knowledge graph to find supporting evidential paths for a given statement (Shiralkar et al., 2017; Syed et al., 2019; Ortona et al., 2018; Shi and Weninger, 2016). The latter approaches are more relevant to our approach presented herein.

There are mainly three sub-types of work using a knowledge graph for truth scoring: (2-1) The first type of works uses only positive evidential paths for truth scoring. For example, the factual statement (`Leonardo da Vinci`, `knownFor`, `Mona Lisa`) will be truthful if there is the supporting positive evidential fact (`Leonardo da Vinci`, `author`, `Mona Lisa`) in a knowledge graph. (Shiralkar et al., 2017) proposed a network flow-based unsupervised approach, called Knowledge Stream (KStream) and Relational Knowledge Linker (KLinker), which use a stream of knowledge to find positive evidential paths supporting that a given statement is true. (Syed et al., 2019) proposed a meta path-based unsupervised approach, called COPPAL, which uses a corroborative meta path to find the positive evidential paths supporting that a given statement is true. Although positive evidence is a good clue for determining truthfulness of a statement, in some cases, negative evidence is necessary because of the incompleteness of a knowledge graph. For example, if there are the facts (`Barack Obama`, `birthYear`, `1961`) and (`Ann Dunham`, `birthYear`, `1942`) in a knowledge graph and the knowledge graph does not have any more information about the two people, we can only determine the given statement (`Barack Obama`, `child`, `Ann Dunham`) as false by the negative evidential path (`Barack Obama`, `birthYear`, `1961`) $\wedge$ (`1961`, $>$, `1942`) $\wedge$ (`Ann Dunham`, `birthYear`, `1942`) that means "*The birth year of Barack Obama is later than that of Ann Dunham*", which implies Ann Dunham cannot be a child of Barack Obama. On the other side, (2-2) the second type of works uses only negative evidential paths for truth scoring. This type of approach suffers from the afore-mentioned same issue with the first type of approach. (Ortona et al., 2018) proposed a rule-based unsupervised approach, called RUDIK, which uses negative rules to find the negative evidential paths supporting that a given statement is false. RUDIK uses a negative sampling based on Extended Local Closed World Assumption (E-LCWA) for

generating negative examples used for learning negative rules. The E-LCWA-based negative sampling suffers from generating false negatives that are actually not false but true. For example, at least 47.54% of the negative examples for relatives generated by the E-LCWA-based negative sampling are actually true according to our analysis. (2-3) The last type of works uses both types of evidential paths for truth scoring. This type of approach suffers less from the afore-mentioned issue than the first and second type of approach. Our ablation study shows that using both types of evidence is more effective for a truth scoring task by up to 0.25 AUC-ROC than only using a single type of evidence. (Shi and Weninger, 2016) proposed a predicate path-based supervised approach, called PredPath, which uses a discriminative predicate path to find positive and negative evidential paths in a knowledge graph supporting that a given statement is true or false. PredPath weights a discriminative predicate path by only considering the correct examples to be covered by the path and ignoring counter examples not to be covered by the path. A discriminative predicate path they proposed is a kind of rule, and a rule can only be properly weighted by considering whether a rule covers correct examples as well as considering whether a rule does not cover counter examples. Our ablation study shows that the counter-weight for a rule, which considers correct examples as well as counter examples is more effective for a truth scoring task by up to 0.2 AUC-ROC than the weight for a rule which only considers correct examples.

3 Problem Statement

In this paper, we address the following problem: Given a knowledge graph G and a factual statement (s, p, o), compute a truth score ranging from 0.0 (untruthful) to 1.0 (truthful) for the given factual statement. A truth score gets closer to 1.0 if the given statement becomes more truthful. Our approach finds positive evidential paths supporting a given statement is true as well as negative evidential paths supporting a given statement is false to calculate a truth score for a given factual statement. To find such evidential paths, our approach uses positive and negative rules. A positive rule comprises a positive evidential path in a body part and a concluding statement in a head part. For example, (x, nationality, y) ← (x, birthplace, z) ∧ (z, country, y) is the positive rule which means the statement (x, nationality, y) is truthful if there is the positive evidential path (x, birthplace, z) ∧ (z, country, y) in a knowledge graph. A negative rule comprises a negative evidential path in a body part and a concluding statement in a head part. For example, ¬(x, nationality, y) ← (x, nationality, z) ∧ (z, ≠, y) is the negative rule that means the statement (x, nationality, y) is untruthful if there is the negative evidential path (x, nationality, z) ∧ (z, ≠, y) in a knowledge graph.

4 Proposed Approach

Our approach is a pipeline that mainly consists of three steps: (1) generation of examples for training, (2) learning positive and negative rules, (3) evidence finding and truth scoring. Figure 1 shows the overall workflow of the proposed approach. In the first step, we generate positive and negative examples used for learning positive and negative rules. A positive example is a true factual statement and a negative example is a false factual statement. As a knowledge graph is a set of true statements, we use a knowledge graph itself as positive examples for training. On the other hand, in most cases, a knowledge graph is not intended to contain false statements. Therefore, we generate false statements from true statements in a knowledge graph by negative sampling and use them as negative examples for training. In the following step, positive and negative rules are learned by using the examples generated in the first step. When learning positive rules, positive examples are used as correct examples and negative examples are used as counter examples. On the contrary, when learning a negative rule, negative examples are used as correct examples and positive examples are used as counter examples. The rule learning step generates a set of rules from input correct examples and weights each of the generated rules by the evidential strength of paths found by the rules. For example, the rule (x, country, y) ← (x, nationality, y) has more strong evidential power than the rule (x, nationality, y) ← (x, resident, y). In the last step, given a factual statement to validate, we find positive and negative evidential paths in a knowledge graph for the given statement by learned positive and negative rules, and calculate a final truth score by unsupervised ensemble of the found positive and negative evidential paths.

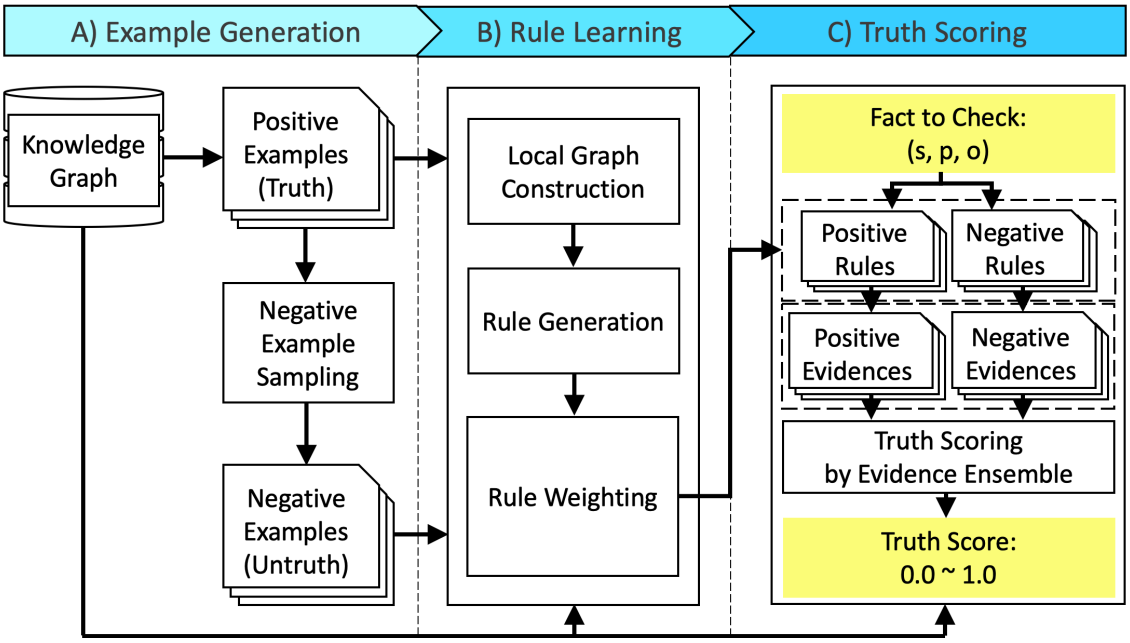

Figure 1: The overall workflow of the proposed approach.

4.1 Negative Example Sampling

A negative example is a false statement, and is necessary for generating negative rules. In most cases, negative examples are not contained in a knowledge graph. Therefore, we have to generate them from the existing positive examples by a negative sampling method. We present a novel negative sampling method based on Distant Local Closed World Assumption (D-LCWA). D-LCWA states that if a knowledge graph contains one or more object values for a given subject and predicate, then there may be other possible values adjacent to the given subject and no other possible values far from the given subject. For example, if a knowledge graph contains one or more children of Clint Eastwood, then it may contain possible children adjacent to him and may not contain possible children far from him.

The D-LCWA-based negative sampling generates a false statement as follows: given a true statement (s, p, o), generate a false statement (s, p, o') by replacing o with o' which has the same type with o and has a distance of two or more and $maxPathLen$ or less from s (i.e., a distance constraint). For example, given the true statement (`Clint Eastwood`, `child`, `Alison Eastwood`), we can generate (`Clint Eastwood`, `child`, `Laura Linney`) based on D-LCWA when Laura Linney has the same type with Alison Eastwood and has a distance of two from Clint Eastwood in a knowledge graph.

D-LCWA vs. E-LCWA. (Ortona et al., 2018) proposed the negative sampling method based on Extended Local Closed World Assumption (E-LCWA), which generates a false statement (s, p, o') by replacing o with o' that has the same type with o and is adjacent to s in a true statement (s, p, o). In our observation, using o' adjacent to s for the replacement often causes false negatives. For example, at least 47.54% of the negative examples for relatives generated by the E-LCWA-based sampling are actually true according to our analysis. To prevent this issue, the D-LCWA-based negative sampling uses the distance constraint that does not consider o' adjacent to s to generate negative examples.

4.2 Rule Generation

We generate a set of rules by using generated examples as follows: (1) We first construct a local graph for each generated example, which is used for generating rules. A *local graph* for an example (s, p, o) is a subgraph that comprises a set of paths between s and o, which have a length $maxPathLen$ or less. To enhance the expressive power of generated rules, we add the $\neq$ predicate to a local graph by comparing all combinations of object-type entities contained in a local graph, such as person, location, organization, and so on, and add the $>$ and $<$ predicate to a local graph by comparing all combinations of data-type entities contained in a local graph, such as integer, real number, and datetime. (2) After constructing local graphs for all examples, we generate an instance of a rule from a constructed local graph.

Specifically, given an example (s, p, o), we generate an instance of a rule whose head part is (s, p, o) and body part is a path between s and o in a local graph, which has a length $maxPathLen$ or less. (3) Finally, we generate a rule by replacing all entities contained in an instance of a rule with corresponding variables. By the afore-mentioned three steps, we generate positive rules by using generated positive examples, and generate negative rules by using generated negative examples.

4.3 Rule Weighting

A weight of a rule is a value from 0.0 (strong) to 1.0 (weak) to express the evidential strength of the given rule. For example, the rule (x, child, y) ← (y, parent, x) is an evidentially *strong rule* because the rule covers all correct examples and does not cover any counter examples. For another example, the rule (x, child, y) ← (x, residence, z) ∧ (y, residence, z) is an evidentially *weak rule* because all people lived with someone are not children of him or her. A weight of a rule gets closer to 0.0 if the given rule has the ability to find stronger evidential paths. In this paper, we consider both correct and counter examples to calculate weights of generated rules. We calculate a weight of a rule, $w_2(r)$, which considers correct examples, $E_{correct}$, as well as counter examples, $E_{counter}$, as follows:

$$w_2(r) = \alpha \times \left(1 - \frac{|C_r(E_{correct})|}{|U_r(E_{correct})|}\right) + \beta \times \frac{|C_r(E_{counter})|}{|U_r(E_{counter})|} \quad (1)$$

where $C_r(E)$ is the number of examples in E covered by r, $U_r(E)$ is the number of examples in E covered by unbounded[2] r, and α and β are the real constants that become 1.0 when added. The stronger a weight of a rule, the more correct and less counter examples the rule covers. This weight measure works in most cases, but not in the case that a rule covers the small number of correct and counter examples. For example, given the weak rule (x, spouse, y) ← (x, occupation, z) ∧ (y, occupation, z) which covers 5% of correct examples and 3% of counter examples, the weight, w_2, of the rule becomes 0.12 which is the fairly strong weight close to 0.0. It is certain that the weight should not be overestimated over the evidential strength of a given rule. To solve this issue, we present a novel counter-weight of a rule, $w_c(r)$, as follows:

$$w_c(r) = 1 - \frac{C_r(E_{correct}) \; - \; 1/\gamma \; \times \; C_r(E_{counter})}{C_r(E_{correct})}\big(1 - w_2(r)\big) \quad (2)$$

where γ is a real constant ranging from 0.0 to 1.0, which is to adjust the impact of $C_r(E_{counter})$. This weight has the constraint that the overall weight value is lowered as much as $C_r(E_{counter})$ is greater than or similar with $C_r(E_{correct})$, which prevents weak rules from being strongly weighted.

Weighting by both types of examples vs. Weighting by only correct examples. Most existing fact checking approaches, KStream, KLinker, COPPAL, and PredPath, only uses correct examples to calculate a weight of a rule. In our observation, rule weighting without considering counter examples often makes an inappropriate situation in which a weak rule is strongly weighted. For example, the weak rule (x, child, y) ← (x, nationality, z) ∧ (y, nationality, z) which covers 76% of correct examples will be strongly weighted even though the rule covers 46% of counter examples. To prevent this issue, the proposed weight uses the constraint to weaken a weight as much as covering counter examples. According to our ablation study, the weight considering both types of examples is more effective for a truth scoring task by up to 0.2 AUC-ROC than the weight only considering correct examples.

4.4 Evidence Finding and Truth Scoring

Given a factual statement to validate, we find positive and negative evidential paths for the given factual statement by learned rules. To find evidential paths, we check whether a learned rule covers a given factual statement or not. If a rule covers a factual statement, it indicates that there is an evidential path

[2] An unbounded rule is obtained by replacing variables paired with x and y in the body part of a rule to new unique variables. For example, given the rule (x, child, y) ← (x, residence, z) ∧ (y, residence, z), the unbounded rule (x, child, y) ← (x, residence, a) ∧ (y, residence, b) is obtained by replacing the variable z to the new unique variables, a and b.

for the body part of the rule in a knowledge graph. For example, if the statement (`Barack Obama`, `child`, `Sasha Obama`) is covered by the rule (`x`, `child`, `y`) ← (`y`, `parent`, `x`), then it indicates that there is the evidential path (`Sasha Obama`, `parent`, `Barack Obama`) in a knowledge graph.

We calculate a truth score for a given factual statement, $\mathcal{S}(s,p,o)$, by unsupervised ensemble of counter-weighted positive and negative rules covering the given factual statement as follows:

$$\mathcal{S}(s,p,o) = \Big(\big(1 - \mathrm{w_c}(r_p)\big) - \big(1 - \mathrm{w_c}(r_n)\big) + 1\Big)/2 \tag{3}$$

where r_p is the strongest positive rule that covers (s,p,o) and r_n is the strongest negative rule that covers (s,p,o). If positive evidence for a given statement are found in a knowledge graph, then we use $\mathrm{w_c}(r_n)$ as 0.0 because it shows better performance averagely 0.01 AUC-ROC than not. Our evidence ensemble approach is totally unsupervised and does not require any human-labeled training data while outperforming the state-of-the-art unsupervised and supervised approaches in the three different datasets.

5 Experiments

5.1 Evaluation Dataset

For evaluation, we use the publicly available standard datasets[3] as well as our dataset. The standard datasets used are as follows: The *Synthetic* dataset constructed by (Shiralkar et al., 2017), which mainly comprises true statements manually extracted from Wikipedia tables and false statements generated by the LCWA-based negative sampling. The *Real-World* dataset derived from Google Relation Extraction Corpora[4] and WSDM Cup Triple Scoring Challenge[5], which mainly comprises true statements manually extracted from Wikipedia texts and false statements generated by the LCWA-based negative sampling.

There are some common issues in the afore-mentioned standard datasets: (1) False-labeled statements in the datasets are automatically generated by the LCWA-based negative sampling. The LCWA-based negative sampling has an issue that it can generate false negatives which are actually not false statements but true statements. Therefore, the datasets contain false-labeled true statements. According to our analysis, at least 4% of false-labeled statements in the datasets are contained in the existing knowledge graph, DBpedia (Auer et al, 2007), which means that they are actually true statements. For accurate evaluation, we corrected such false negatives by re-labeling them as true and used the corrected datasets in our evaluation. (2) Some true-labeled statements in the datasets are actually known facts already contained in the existing knowledge graph, DBpedia, which means the test cases in the datasets can be easily solved by checking whether a given statement is contained in a knowledge graph or not. This is not suitable for our goal that is to validate unknown facts missing in a knowledge graph. According to our analysis, 77.26% of true-labeled statements in the Synthetic dataset are contained in DBpedia. In the case of the Real-World dataset 8.58% of true-labeled statements are contained in DBpedia, which seems small, but for some predicates, e.g., nationality and profession, 100% of true-labeled statements are known facts already contained in DBpedia.

We present a novel evaluation dataset for a fact checking problem, which is constructed by manually labeling true or false label to each of the factual statements extracted from Wikipedia texts by the state-of-the-art BERT-based information extractor (Nam et al., 2020). Our dataset is built to satisfy our goal in two ways: (1) All the false-labeled statements in our dataset are manually validated to prevent from inaccurate evaluation by false negatives. (2) The true-labeled statements are included in our dataset only if a given statement is not a known fact contained in the existing knowledge graph, K-Box (Nam et al., 2018); It makes our dataset more challenging in a fact checking problem than the other standard datasets. The statistics of the three datasets are shown in Table 1.

[3] https://github.com/shiralkarprashant/knowledgestream
[4] https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html
[5] https://www.wsdm-cup-2017.org/triple-scoring.html

Dataset	Predicate #	Factual Statement			True-False Ratio
		True #	False #	Total #	
Synthetic	8	839	7253	8092	10 : 90
Real-World	6	7565	22688	30253	25 : 75
Ours	35	290	1469	1759	16 : 84

Table 1: Statistics of the Synthetic, Real-World, and our datasets.

5.2 Experiment Settings

Knowledge Graph. For evaluation on the Synthetic and Real-World datasets, we use English DBpedia as a background knowledge graph, and for evaluation on our dataset, we use K-Box as a background knowledge graph. We use the AUC-ROC score as an evaluation metric and set $\alpha = 0.1$, $\beta = 0.9$, and $\gamma = 0.25$ as the parameters of our approach, which shows the best performance in our experiment.

Competing Approaches. We compare our approach to the state-of-the-art fact checking approaches: (1) KStream, (2) KLinker, (3) COPPAL, (4) RUDIK, and (5) PredPath. The difference between the approaches is shown in Table 2. KStream, KLinker, COPPAL, and RUDIK are the unsupervised approaches that use a single type of evidence in truth scoring. PredPath is the supervised approach that uses both types of evidence in truth scoring. RUDIK weight rules by both types of examples. The other competing approaches weight rules only by correct examples. For KStream, KLinker, and PredPath, we used the implementation provided by their authors[6]. For COPPAL, we used the implementation published at the repository[7]. For RUDIK, we reproduced the implementation based on the paper published by (Ortona et al., 2018). For all the approaches, we used the parameter settings published by their authors.

	Competing Approaches					Ours
	KStream	KLinker	COPPAL	RUDIK	PredPath	
Learning Type	U	U	U	U	S	U
Evidence Type	P	P	P	N	P & N	P & N
Negative Sampling	-	-	-	E-LCWA	Human	D-LCWA
Evidence Weighting	W1	W1	W1	W2-M	W1	W2-C

Table 2: Difference between approaches where U denotes unsupervised learning, S denotes supervised learning, P and N denote positive and negative evidence each, W1 denotes a weight only considering correct examples, W2 denotes a weight considering both correct and counter examples, W2-M is the marginal weight proposed in RUDIK, and W2-C is the counter-weight proposed in this paper.

5.3 Comparison Result

Table 3 and 4 show the comparison results in the two standard datasets. (1) The positive evidence-based approaches, KStream, KLinker, and COPPAL, show the better performance by up to 32.01% than the negative evidence-based approach, RUDIK. (2) The supervised approach, PredPath, shows better performance by up to 17.34% than the unsupervised positive evidence-based approaches. (3) Our unsupervised approach outperforms the supervised approach, PredPath, by up to 5.57%.

The order of performance is similar in our dataset, which is shown in Table 5 and Figure 2. (1) The positive evidence-based approaches show the better performance by up to 11.15% than the negative evidence-based approach. (2) The supervised approach shows the better performance by up to 13.71% than the unsupervised positive evidence-based approaches. (3) Our unsupervised approach outperforms the supervised approach by up to 3.72%.

The main difference between our unsupervised approach and PredPath is a rule weighting measure. PredPath weights the evidential strength of a rule by using only correct examples while our approach weights the evidential strength of a rule by using correct and counter examples. Our ablation study in Section 4.3 supports that this can make a huge difference in performance in a truth scoring task.

[6] https://github.com/shiralkarprashant/knowledgestream
[7] https://github.com/dice-group/COPAAL

Model	Predicates in the Synthetic dataset								Total
	battle	birth-Place	capital	direc-tor	key-Person	spouse	team	vicePre sident	
KStream	0.9765	0.0469	0.9979	0.9796	0.7975	0.9953	0.9639	0.9954	0.7717
KLinker	0.9684	0.0269	0.9936	0.9683	0.7956	0.9759	0.9706	0.9868	0.8002
COPPAL	0.927	0.1768	0.9906	0.7594	0.8312	0.9805	0.8826	0.8307	0.833
RUDIK	0.411	0.7108	0.3958	0.5542	0.5404	0.9203	0.4992	0.6006	0.5464
PredPath	0.9401	0.9545	**1.0**	0.9808	0.8714	**1.0**	0.9535	**1.0**	0.9451
Ours	**1.0**	**1.0**	0.9807	**0.9977**	**0.9215**	**1.0**	**0.9746**	0.9985	**0.9598**

Table 3: AUC-ROC performance scores in the Synthetic dataset.

Model	Predicates in the Real-World dataset						Total
	almaMa-ter	birthPlace	death-Place	education	national-ity	profession	
KStream	0.7885	0.7414	0.7859	0.7454	0.938	0.9474	0.769
KLinker	0.8064	0.8315	0.8135	0.7734	0.9673	0.9281	0.785
COPPAL	0.6502	0.7292	0.7088	0.5011	0.9716	0.8553	0.6494
RUDIK	0.459	0.5167	0.5358	0.4891	0.4686	0.5	0.4649
PredPath	0.7242	0.7943	0.7632	**0.8335**	**1.0**	0.8227	0.7374
Ours	**0.8071**	**0.8715**	**0.8332**	0.7532	**1.0**	**1.0**	**0.7931**

Table 4: AUC-ROC performance scores in the Real-World dataset.

	KStream	KLinker	COPPAL	RUDIK	PredPath	Ours-dben	Ours-kbox
Total	0.6372	0.6166	0.5864	0.5257	0.6628	0.6884	**0.7**

Table 5: Total AUC-ROC performance scores in our dataset where *Ours-dben* and *Ours-kbox* are the proposed model whose parameters are tuned on English DBpedia and K-Box each.

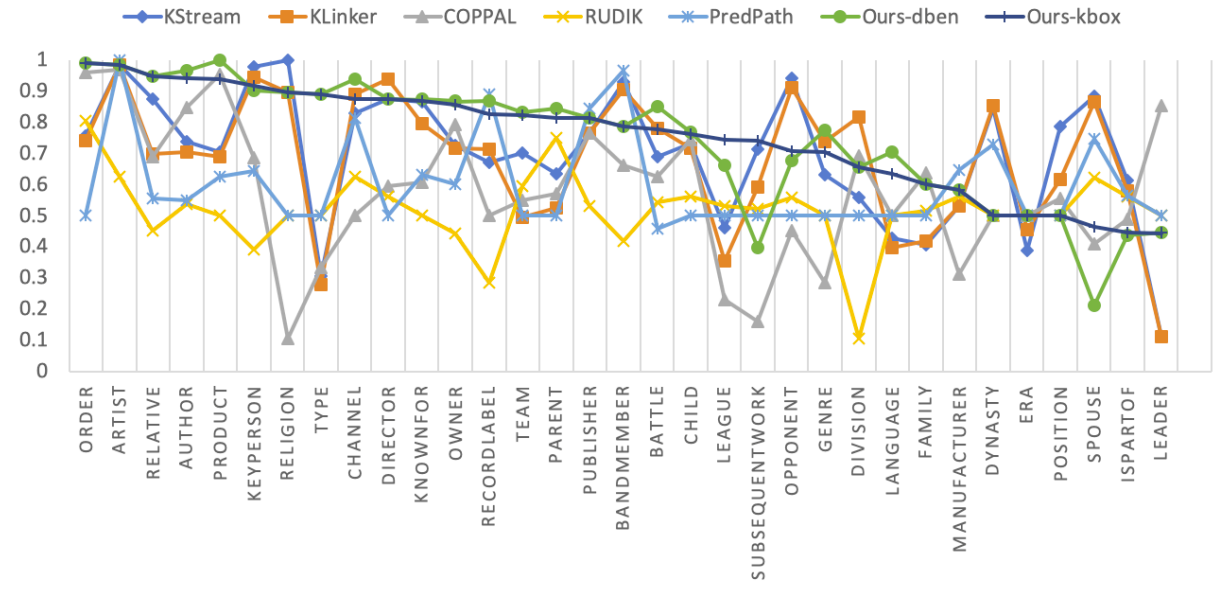

Figure 2: Detailed AUC-ROC performance scores in our dataset. The *x* axis indicates a predicate in our dataset and the *y* axis indicates AUC-ROC scores for each predicate.

5.4 Ablation Study

To see how each of the proposed methods contributes to the overall performance in a truth scoring task, we conducted an ablation study, and the result is shown in Table 4. The D-LCWA-based negative sampling is more effective by up to 0.0509 average AUC-ROC than the existing methods based on LCWA and E-LCWA. The good performance is because of the fact that the D-LCWA-based method uses the distance constraints to reduce generation of false negatives. The LCWA and E-LCWA-based methods have no such constraints, which makes them suffer from the performance drop by false negatives.

As for the rule weighting task, the counter-weight, W2-C, is more effective by up to 0.0785 average AUC-ROC than the other methods. W2-C uses both correct and counter examples to weight the evidential strength of a rule. On the contrary, W1 only uses correct examples. These facts make our proposed weight 7% (AUC-ROC) more effective than W1, which supports the reason why our unsupervised fact checking approach using W2-C outperforms the supervised approach, PredPath, using the weight measure which is a kind of W1.

As for the truth scoring task, using positive and negative evidence (P & N) is improved slightly by 0.0037 average AUC-ROC compared to using only positive evidence (P), and is hugely improved by 0.2512 average AUC-ROC compared to using only negative evidence (N). The point is that using both types of evidence shows the best performance on all the datasets, which indicates that positive and negative evidence are complement each other to solve other types of test cases in a fact checking problem.

Task	Method	Truth Scoring Performance			Average
		Synthetic	Real-World	Ours	
Negative Sampling	LCWA	0.9131	**0.8214**	0.6831	0.8059
	E-LCWA	0.8988	0.7891	0.6123	0.7667
	D-LCWA	**0.9598**	0.7931	**0.7**	**0.8176**
Rule Weighting	W1	0.9115	0.7408	0.5766	0.743
	W2	0.9464	**0.7958**	0.6908	0.811
	W2-M	0.9423	0.5897	0.6853	0.7391
	W2-C	**0.9598**	0.7931	**0.7**	**0.8176**
Truth Scoring	P	0.9597	0.7838	0.6983	0.8139
	N	0.8042	0.3665	0.5284	0.5664
	P & N	**0.9598**	**0.7931**	**0.7**	**0.8176**

Table 4: Results of the ablation study where W1 is the weight only considering correct examples, W2 is the weight considering both correct and counter examples, W2-M is the marginal weight used by RUDIK, and W2-C is the counter-weight used by our approach. P is to use positive evidence, N is to use negative evidence, and P & N is to use both types of evidence to calculate a truth score.

6 Conclusion

We presented a rule-based unsupervised approach for a truth scoring task in a knowledge graph, based on 1) unsupervised ensemble of positive and negative evidence found by 2) positive and negative rules which are learned from the learning examples generated by 3) D-LCWA-based negative sampling and are weighted by 4) the counter-weight considering both correct and counter examples. We validated our approach on the fact checking dataset first presented in this paper as well as on the two different standard datasets. The result showed that our unsupervised approach significantly outperforms the state-of-the-art unsupervised approaches by up to 12.68% (AUC-ROC) and even outperforms the supervised approach by up to 5.57% (AUC-ROC) in the three different datasets. Our approach is fully unsupervised and can be easily extended to a wide range of predicates in a knowledge graph.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2013-2-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform). This work was supported by Institute of

Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [2016-0-00562(R0124-16-0002), Emotional Intelligence Technology to Infer Human Emotion and Carry on Dialogue Accordingly].

Reference

Baoxu Shi, and Tim Weninger. "Discriminative predicate path mining for fact checking in knowledge graphs." Knowledge-based systems 104 (2016): 123-133.

Bonan Min, Marjorie Freedman, and Talya Meltzer. "Probabilistic inference for cold start knowledge base population with prior world knowledge." In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 601-612. 2017.

Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. "Defacto—temporal and multilingual deep fact validation." Journal of Web Semantics, 35: 85-101, 2015.

James Thorne and Andreas Vlachos. "Automated Fact Checking: Task Formulations, Methods and Future Directions." Proceedings of the 27th International Conference on Computational Linguistics, pp. 3346–3359. 2018.

Sangha Nam, Eun-kyung Kim, Jiho Kim, Yoosung Jung, Kijong Han, and Key-Sun Choi. "A Korean Knowledge Extraction System for Enriching a KBox." In Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pp. 20-24. 2018.

Sangha Nam, Minho Lee, Donghwan Kim, Kijong Han, Kuntae Kim, Sooji Yoon, Eun-kyung Kim, and Key-Sun Choi. "Effective Crowdsourcing of Multiple Tasks for Comprehensive Knowledge Extraction." In Proceedings of The 12th Language Resources and Evaluation Conference, pp. 212-219. 2020.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "Dbpedia: A nucleus for a web of open data." In The Semantic Web, pp. 722-735. Springer, Berlin, Heidelberg, 2007.

Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. "Robust discovery of positive and negative rules in knowledge bases." In 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1168-1179. IEEE, 2018.

Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. "Finding streams in knowledge graphs to support fact checking." In 2017 IEEE International Conference on Data Mining (ICDM), pp. 859-864. IEEE, 2017.

Zafar Habeeb Syed, Michael Röder, and Axel Cyrille Ngonga Ngomo. "Factcheck: Validating rdf triples using textual evidence." In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1599-1602. 2018.

Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. "Unsupervised discovery of corroborative paths for fact validation." In International Semantic Web Conference, pp. 630-646. Springer, Cham, 2019.