Identifying Surprising Facts in Wikidata

Nicholas Klein[1], Filip Ilievski[1], Hayden Freedman[2] and Pedro Szekely[1]

[1]*Information Sciences Institute, University of Southern California*
[2]*University of California Irvine*

Abstract

To expand their boundary of knowledge, to solve a certain task, or merely to entertain themselves, people on the Web often hunt for surprising facts. While large repositories of structured knowledge like Wikidata hold the promise to provide surprising facts to their users and developers, currently no mechanism exists to identify surprising facts in Wikidata. In this paper, we study the ability of popular embedding models to estimate the surprise level of a fact in Wikidata. We formulate a novel task of *Surprising Fact Identification*, and we create two associated benchmarks: *Trivia* and *Survey*. We propose two scalable methods based on outlier detection and link prediction to estimate surprise scores for any statement in a graph like Wikidata. We evaluate our methods with various embedding models on the two benchmarks. We perform further analysis of the predictions for outlier and non-outlier facts to investigate to what extent link prediction models regress to the mean.

1. Introduction

Surprise is a measure of how unexpected a certain statement is. For example, while it is expected that Vladimir Putin speaks Russian, it is more surprising that he can speak German too. It is expected that Putin is a politician, but it may be surprising that he has studied law. To expand their boundary of knowledge, to solve a certain task, or merely to entertain themselves, people on the Web often hunt for surprising facts [1]. It is therefore more valuable for users to be presented with a surprising fact rather than a trivial one. Recognizing this phenomenon, popular sites on the Web readily provide contextually relevant facts that are fun or surprising.[1]

The expansion of AI technologies and the vast amount of available knowledge provides an opening for surprising facts to be identified automatically. Prior work by Tsurel et al. [1] has investigated the ability of statistical AI methods to extract surprising facts from Wikipedia. Meanwhile, very large and curated structured knowledge sources like Wikidata [2] have emerged, providing over a billion facts about nearly one hundred million entities. While Wikidata's knowledge is intuitively suitable to support human exploration and AI reasoning, the provided information can easily get overwhelming for both humans and machines. Wikidata contains some information (e.g., ranks and temporal qualifiers) that enables prioritizing entity statements for the same property. It also has the Curious Facts Dashboard [3], which presents potentially incorrect entries for user review. However, this system uses a rule-based reasoner and tends to

Wikidata'22: Wikidata workshop at ISWC 2022
nmklein@usc.edu (N. Klein); ilievski@isi.edu (F. Ilievski); hfreedma@uci.edu (H. Freedman); pszekely@isi.edu (P. Szekely)

[1]E.g., https://www.historyhit.com/facts-about-vladimir-putin/, accessed May 20, 2022.

surface problems such as property constraint violations, rather than facts that are ontologically correct but surprising. The emergence of large structured knowledge graphs and representation learning methods [4, 5] brings up the question: *can we apply state-of-the-art representation learning techniques to effectively identify surprising facts in Wikidata?*

In this paper, we study the ability of popular embedding models, like TransE [4], ComplEx [5], and language models like BERT [6], to identify the surprise level of a fact in Wikidata. We formulate *Surprising Fact Identification*, a novel task where a system has to estimate how surprising a Wikidata fact is. We create two benchmarks for this task: *Trivia*, where systems rank the facts per question according to their surprise, and *Survey*, where the goal is to provide a global ranking of 70 facts. We propose two scalable methods based on embedding models with an ability to estimate surprise score for any statement in a graph like Wikidata. We evaluate a list of embedding model variants to perform Surprising Fact Identification on the two benchmarks. Hypothesizing that embedding-based link prediction models regress to the mean, we also analyze whether their predictions for a given subject entity tend towards the values that are most commonly correct for similar entities.

2. Surprising Fact Identification

2.1. Task Definition

We define *surprise* as a measure of how unusual or unexpected a certain statement is. We formulate the task of *Surprising Fact Identification* as a ranking task. For a given set of facts F, we want to assign a ranking to each of the facts $f \in F$ that corresponds to the relative surprise a human would likely have at learning each f is true. Each fact is of the form $f : (e, [(p_1, o_1), ..., (p_n, o_n)])$, consisting of an entity e that is the subject of the fact and a non-empty list of predicate-object pairs (p_i, o_i). A fact f indicates that each of the triples $< e, p_i, o_i >$ is true for $i \in [1, n]$.

2.2. Benchmarks

We collect two Surprising Fact Identification benchmarks: Wikidata-MC-Trivia-118 and Wikidata-Survey-FunFacts-70.

Wikidata-MC-Trivia-118 This benchmark concretely defines surprise of a fact f as the probability that a human would not guess f to be true. It consists of 24 multiple-choice questions with a total of 118 candidate answers, each having an inferred human surprise score. Most questions have five candidate answers, and questions can have more than one correct answer. The questions are about 22 unique entities: 14 humans, 3 sovereign states, 3 paintings, 1 film, and 1 business. Of the 24 questions, 7 are single-answer (e.g., "What is the birthplace of Boris Johnson") and 6 ask about numeric values (e.g., "How many children does Arnold Schwarzenegger have?"). Each question presents a single entity and property, and each candidate answer is a potential object of that entity-property pair. Thus, all facts in this benchmark consist of a single triple. We collected this benchmark with a quiz administered via Google Forms to a group of 26 researchers who participated voluntarily. A screenshot of the quiz is shown in

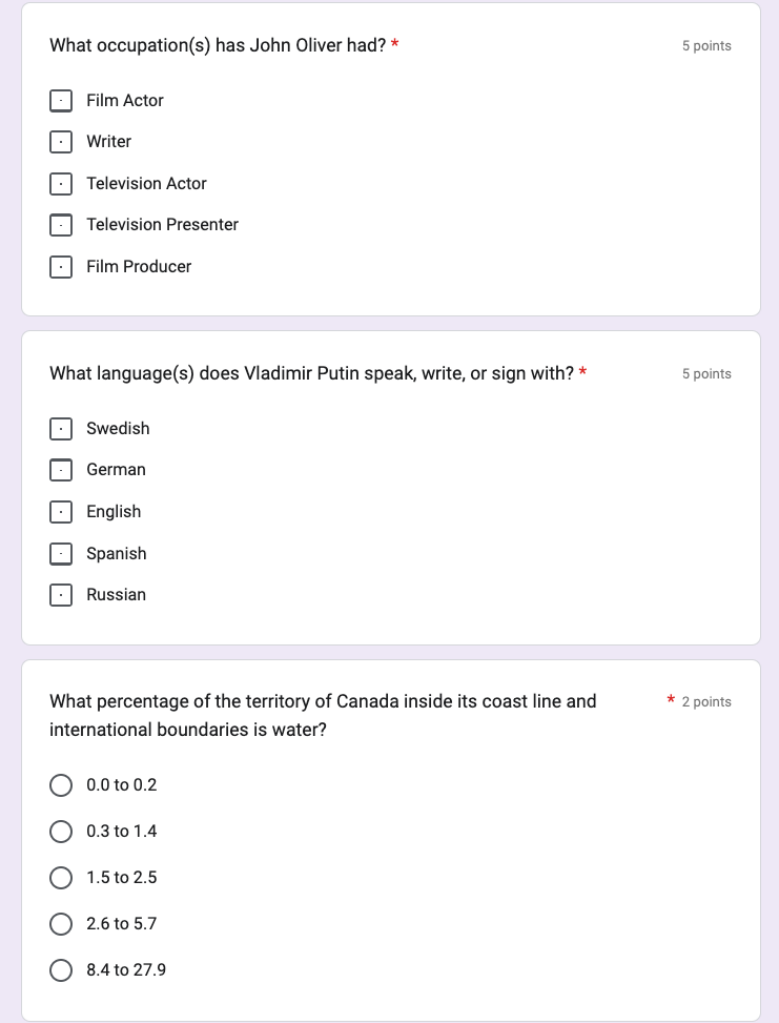

Figure 1: Screenshot of the quiz on Google Forms.

Figure 1. Surprise scores were inferred as the percentage of participants who did not choose each candidate answer.[2]

The quiz was designed to include a uniform distribution of surprising-correct answers, unsurprising-correct answers, and incorrect (distracting) answers. We began by selecting a set of well-known entities from several classes. To help ensure we would find entities that have surprising facts, we did some simple Google searches of the form "<class>s with surprising facts". The set of candidate facts to pose as questions for each entity was refined by utilizing prior work that identifies entities' most salient facts [8]. Next, surprising-correct answers were selected

[2]We expect that if a true statement is considered untrue by many participants, this is indicative of a surprising fact. We expect that statements that are simply unknown would be guessed randomly by participants, leading to an unsurprising score. We do acknowledge that a more principled annotation of this data may need to distinguish between interesting, surprising, and unknown facts [7].

by manual selection of seemingly surprising facts from the bottom half of each entities' facts ordered by frequency of an object for a class-property pair. The selected facts were converted into questions. Facts for the same property were combined into the same question, serving as multiple correct answers. Non-surprising correct answers for the questions were generated by filling in any other answers that are correct per Wikidata. Distracting answers were generated by sampling answers that are incorrect for the entity according to Wikidata, and weighted by the frequency with which they occur for the given property over all entities of the given entity's class. We fact-checked all answers using Google, omitting questions with disputable answers. Finally, we trimmed down the number of candidate answers for each question to a maximum of 5, aiming to keep the distribution even amongst incorrect answers, correct answers expected to be surprising, and correct answers expected to be unsurprising.

Wikidata-Survey-FunFacts-70 consists of 70 facts, each consisting of an entity and a non-empty list of predicate-object pairs. Each fact is labeled with three scores indicating the degree to which humans find the fact *surprising*, think the fact is a *good trivia* question, and would say they *knew* the fact. The facts are about 47 different humans and only include entity-valued properties. This benchmark was created by mapping a subset of the rows from the FunFacts benchmark [1] for Wikipedia. Tsurel et al. [1] collected the facts by several statistical methods and asked crowd workers to judge the degree of fact surprise, fact trivia-worth, and whether they knew the fact before reading it. The scores were then averaged across workers. The original data contains 362 natural language facts about 109 different humans. As the facts in the original dataset are natural language, not all of them can be mapped to Wikidata (e.g., "Einstein offered and was called on to give judgments and opinions on matters often unrelated to theoretical physics or mathematics"). We mapped 113 of these 362 facts, resulting in 70 facts that have been mapped to Wikidata.

2.3. Connection to Prior Studies

Prakash et al. [9] gather trivia facts about Wikipedia entities from IMDb and propose an algorithm to estimate trivia-worthiness of a fact. Tsurel et al. [1] evaluate methods for identification of surprising, trivia-worthy, and unknown facts from articles in Wikipedia. As, to our knowledge, no such work exists for knowledge graphs like Wikidata; we adapt the benchmark introduced in [1] for surprising fact identification in Wikidata. Serban et al. [10] generate natural language questions from Freebase automatically with a neural network, yet, it is unclear if these questions concern surprising facts. Prior work has provided methods for estimating quality of individual Wikidata facts, by estimating aspects such as veracity [11, 12]. Quality estimation and triple classification are orthogonal tasks to ours, and it is unclear how to apply quality estimation methods to identify surprising facts.

Notably, Surprising Facts Identification is related to the popular task of Link Prediction where the goal is to correctly predict the object of a subject-predicate pair. Previous work has explored Link Prediction in Wikidata in various capacities. Safavi et al. [13] perform a comprehensive evaluation of the calibration of several embeddings models. Wu et al. [14] use a rule-based approach to link prediction which is evaluated over several large knowledge bases, including Wikidata. Rosso et al. [15] introduce an embedding-based predictive model that uses graph triples together with their associated key-value pairs to restrict or disambiguate facts

in Wikidata. Joshi et al. [16] propose an embedding-based model that can reduce the search space by automatically selecting a pool of promising entities to reduce computational load. Recognizing this connection and opportunity to build upon prior research, we incorporate link prediction models like TransE [4] and ComplEx [5] into our surprise identification methods, and we perform further analysis of the inherent ability of the models to estimate surprise.

3. Method

We describe two novel methods that score the surprise for a Wikidata fact based on statistical outlier detection and on link prediction.

3.1. Identifying Surprise with Statistical Outlier Detection (SOD)

Our first method measures whether the entity e is an atypical subject (*outlier*) for the property-object pair. We estimate this by comparing e to the set of entities E_f for which $< e', p_i, o_i >$ is true $\forall i \in [1, n]$ and $e' \in E_f : e' \neq e$. For example, to estimate whether $< Putin, language, Russian >$ is an outlier, we would compare Putin to the entities which speak Russian according to Wikidata.

We compute the surprise score for a statement as a ratio between the distance between e and the entities in E_f, and the dispersion of entities within E_f: $Surprise_{outlier}(f) = distance(e, E_f)/dispersion(E_f)$. We utilize embeddings to represent each entity in Wikidata. $distance(e, E_f)$ is a measure of distance from the entity e to the entities in E_f in the embedding space. We experimented with two such formulations: $distance_c(e, E_f)$ computes the cosine distance from e to the centroid of E_f and $distance_{ap}(e, E_f)$ computes the average pairwise cosine distance between e and each $e' \in E_f$. $dispersion(E_f)$ is a measure of how spread out E_f is in the embedding space, where we again experiment with two analogous formulations: $dispersion_c(E_f)$ computes the average cosine distance from each $e' \in E_f$ to the centroid of E_f and $dispersion_{ap}(E_f)$ computes the average pairwise distance between e' and e'' for $e', e'' \in E_f : e' \neq e''$.

Intuitively, the distance term causes an entity e to be found more surprising for a fact f if it is dissimilar from entities that have fact f. The dispersion term normalizes this dissimilarity, decreasing our surprise for facts that many diverse entities have (e.g., "language written, spoken, or signed = English") and increasing our surprise for facts that typically belong to very similar entities (e.g., "Occupation = Basketball Player"). For the edge case where $|E_f| < 2$, there are too few entities in Wikidata that have all predicate-object pairs of f for us to compute a dispersion score, so we assign the f maximum surprise score (∞).

3.2. Identifying Surprise via Link Prediction (LP)

Another approach for identifying surprising facts is by using link prediction. Graph embedding models like TransE and ComplEx define ways of operating on their representations of entities and properties to yield predictions for where in the embedding space the corresponding subject or object entity resides. This gives us two potential methods to compute surprise scores: we define $Surprise_{LP-lhs}(f)$ as the aggregated distance from each $o_i \in f$ to the predicted locations of the objects of $< e, p_i >$, and we define $Surprise_{LP-rhs}(f)$ as the aggregated distances from e to each of the predicted locations of the subjects of $< p_i, o_i >$. There are several additional

settings for these methods: the distances in both the lhs and rhs methods can be aggregated using either *max* or *avg*; the rhs method can be modified to instead compute the distance from e to the centroid of the predicted object locations; and various distance functions can be used here, including cosine, L2, and negative dot-product.

3.3. Experimental Setup

Evaluation protocol We evaluate our methods' ability to identify surprising facts by measuring correlation between their surprise scores with the human scores available in each benchmark using Spearman's ρ and Kendall's τ. For *Wikidata-MC-Trivia-118*, we evaluate our method's surprise scores by measuring their correlation with the inferred human surprise scores. To mimic the format that this benchmark was created in, we measure the average correlation of answers' surprise scores within each question. We additionally report separate results for the subsets of questions that ask about entity-valued and numeric-valued properties. For *Wikidata-Survey-FunFacts-70*, we evaluate our method's surprise scores by measuring their correlation with the crowdsourced scores for *goodTrivia*, *surprise*, and *knew*. A successful surprise identification method will give scores that correlate positively with *goodTrivia* and *surprise*, and negatively with *knew*. As this benchmark was created by presenting each fact to a human annotator independently, we measure the global correlation over all facts.

Models We experiment with eight embedding models: (1) *BERT* text embeddings that we computed by automatically generating a node description based on seven properties and using sentence-transformers to encode the resulting sentence; [3] (2, 3) *TransE* and *ComplEx* graph embeddings computed directly on Wikidata; (4, 5) TransE and ComplEx graph embeddings that have been computed over a graph derived from Wikidata by abstractive summarization into 'profiles' (*P-TransE* and *P-ComplEx*) [8] ; (6, 7, 8) *H*, *A*, and *S* random-walk-based graph embeddings, designed to capture similarity based on homophily, numeric attributes, and structure, respectively; (9) *kNN-P-TransE* supervised TransE embeddings, based on a kNN model for each property.

Method details We evaluate our outlier-based method with the eight unsupervised embedding models. We limit the size of E_f for each fact to 10,000 by sampling. We evaluate our LP method with the embedding models *TransE*, *ComplEx*, *P-TransE*, *P-ComplEx*, and *kNN-P-TransE*, as these allow for unsupervised link prediction via translation and complex-diagonal operators. Vanilla *TransE* and *ComplEx* are not trained on numeric-valued edges and therefore cannot be used directly for LP on numeric-valued properties. Because the P- variants were trained on a graph that replaces numeric values with nodes corresponding to ranges, they *can* be used for LP on numeric-valued properties. Meanwhile, as the profile graph omits information from the original graph, the profile (P-) embeddings lack information for 19 of the facts in the Wikidata-Survey-FunFacts-70 dataset, while the kNN model lacks information for 10 facts. For these missing facts, we use a frequency method as a fallback strategy. For the Wikidata-MC-Trivia-118 benchmark, we report $Surprise_{LP-lhs}$ for the vanilla and the supervised graph embeddings, and $Surprise_{LP-rhs}$ for the profile-graph embeddings. For the Wikidata-Survey-FunFacts-70 benchmark, we use $Surprise_{LP-rhs}$ for the unsupervised LP models, aggregating predictions by

[3] Properties used: P31 (instance of), P279 (subclass of), P106 (occupation), P39 (position held), P1382 (partially coincident with), P373 (Commons Category), and P452 (industry).

Table 1
Results on Wikidata-MC-Trivia-118.

Methods		Qnode facts		Numeric facts		All facts	
		Rho	**Tau**	**Rho**	**Tau**	**Rho**	**Tau**
Baselines	**random**	-0.003	-0.002	0.024	0.019	0.003	0.003
	frequency	0.043	0.055	0.134	0.129	0.066	0.074
SOD	**BERT**	0.574	0.502	**0.49**	**0.394**	0.553	**0.475**
	ComplEx	0.556	0.468	0.455	0.382	0.531	0.447
	TransE	**0.638**	**0.526**	0.326	0.228	**0.56**	0.452
	P-ComplEx	0.429	0.381	0.056	0.075	0.335	0.305
	P-Transe	0.421	0.382	0.211	0.137	0.368	0.32
	H-RandomWalk	0.594	0.505	0.401	0.348	0.546	0.466
	A-RandomWalk	0.081	0.088	0.171	0.152	0.103	0.104
	S-RandomWalk	-0.025	-0.067	0.381	0.286	0.076	0.021
LP	**ComplEx**	0.475	0.413	-	-	-	-
	TransE	0.442	0.391	-	-	-	-
	P-ComplEx	0.392	0.346	0.219	0.18	0.348	0.304
	P-Transe	0.246	0.211	0.422	0.376	0.29	0.252
	kNN-P-TransE	0.109	0.099	0.639	0.565	0.242	0.215

measuring the distance from their centroid to the given fact's subject entity. Analogously, we use $Surprise_{LP-lhs}$ for the supervised LP model. We use cosine distance for all LP methods.

We define two baselines: *random* and *frequency*. The frequency-based method assigns the surprise score for a fact as the percent of Wikidata entities that it does not apply to.

4. Results

The results on both benchmarks in tables 1 and 2 show that our SOD methods are able to identify surprising facts better than the baselines and our LP methods. On the Trivia benchmark, SOD yields the highest correlation scores with the BERT, TransE, and H embeddings, followed closely by the ComplEx embeddings. On the Survey benchmark, the H-RandomWalk embeddings perform the best on all three crowd-sourced scores, followed by ComplEx and TransE on the "goodTrivia" and "surprising" scores, and S-RandomWalk and BERT on "Knew". The performance of the baselines is low on the Trivia benchmark, while the frequency baseline performs notably better on the Survey benchmark, giving scores that are competitive with most of our models. We think that this is due to the presence of compound facts in the Survey benchmark, whose combination reveals unlikely facts that align with human judgments of trivia-worthiness or surprise. Yet, several of the SOD methods improve over the frequency baseline on this benchmark as well, revealing that the subject entity embeddings provide key additional information that is not captured by the predicate-object frequencies.

Curiously, link prediction with the same embedding models performs consistently worse than the SOD methods overall and on the entity-valued facts, though this trend is reversed for the P-TransE and P-ComplEx on the numeric facts. This shows that, while the embedding models may contain valuable information, the method that uses them to identify surprising facts

Table 2
Results on Wikidata-Survey-FunFacts-70. The P-ComplEx and P-TransE models with the LP method lack scores for 19 of the 70 facts, and the kNN-P-TransE model lacks scores for 10 facts. For those facts, we fall back to the frequency baseline to fill in a surprise score. We mark these results with an asterisk (*).

Methods		GoodTrivia (↑)		Surprising (↑)		Knew (↓)	
		Rho	**Tau**	**Rho**	**Tau**	**Rho**	**Tau**
Baselines	**random**	-0.005	-0.003	-0.005	-0.004	0.005	0.004
	frequency	0.328	0.236	0.329	0.238	-0.410	-0.304
SOD	**BERT**	0.396	0.271	0.345	0.235	-0.307	-0.218
	ComplEx	0.478	0.345	0.373	0.261	-0.247	-0.177
	TransE	0.460	0.314	0.350	0.237	-0.244	-0.180
	P-ComplEx	0.230	0.160	0.137	0.094	-0.074	-0.057
	P-Transe	0.353	0.250	0.257	0.178	-0.144	-0.107
	H-RandomWalk	**0.540**	**0.355**	**0.521**	**0.359**	**-0.499**	**-0.366**
	A-RandomWalk	0.311	0.224	0.240	0.169	-0.199	-0.110
	S-RandomWalk	0.331	0.229	0.255	0.183	-0.333	-0.243
LP	**ComplEx**	0.146	0.113	0.148	0.111	-0.170	-0.126
	TransE	0.197	0.141	0.162	0.114	-0.109	-0.069
	P-ComplEx	0.086*	0.058*	0.018*	0.009*	0.087*	0.056*
	P-Transe	0.183*	0.145*	0.174*	0.131*	-0.080*	-0.050*
	kNN-P-TransE	0.076*	0.061*	0.048*	0.031*	-0.073*	-0.060*

Table 3
MRR results on the Trivia dataset for TransE-based Link Prediction models. We show MRR results for the entire benchmark, only on the correct/incorrect triples, and only on the (non-)outliers subsets. For each question, we take the two facts closest to the centroid to be non-outliers, and the two facts furthest from the centroid to be outliers.

LP Model	TransE	P-TransE	kNN-P-TransE
All	0.0009	0.0006	0.2213
Correct	0.0012	0.0004	0.2617
Incorrect	0.0005	0.0008	0.1787
Non-Outliers	0.0011	0.0005	0.2752
Outliers	0.0007	0.0007	0.1493

plays a key role in the final performance. While LP performs consistently better than random on both benchmarks, it has a relatively poor performance on the Survey benchmark, which may again be attributed to the presence of compound facts in this benchmark. Link prediction predicts a single subject vector at a time for each predicate-object pair and we account for the compound facts afterwards by computing their centroid. The centroid of these predicted vectors is an estimate of where in the embedding space an entity that has all of the predicate object pairs would be. This estimate may be poor when the predicted subject vectors are far apart (as in the case of facts that combine predicate-object pairs that rarely co-occur), and thus the LP method cannot as directly and effectively take into consideration interactions between such predicate-object pairs that are important to humans for determining if a fact is surprising.

Hypothesizing that LP methods regress to the mean of the entities with similar facts, we study the behavior of the embedding models further by analyzing the LP MRR of a fact in relation to the centroid of the cluster. The results are shown in table 3. The MRR for two of the three models is higher on the correct subset compared to the incorrect one, and on the non-outlier facts compared to the outliers. We observe that across any subset of the data, the supervised embedding model obtains highest MRR. Interestingly, while the supervised model performs the best on the link prediction task, its performance on the surprise task is the lowest out of the three models on this benchmark. The unsupervised version, P-TransE, assigns a comparably low MRR value on both the outliers and the non-outliers, yet, its performance on the surprise task is notably higher. This means that LP models optimize to predict a value that is typical rather than surprising. Based on these results, we hypothesize that more accurate link prediction models may not necessarily be better at predicting surprising facts; subsequent research on this topic is beneficial to further analyze their behavior.

5. Conclusions

In this paper, we studied the challenge of identifying surprising facts in Wikidata automatically. Inspired by earlier work on Wikipedia, we formulated a novel task of *Surprising Fact Identification* where an AI system has to mimic the surprise estimation of humans. We developed two novel benchmarks to evaluate representation learning models on this task. We proposed two generic methods to identify surprising facts based on statistical outlier detection and link prediction. Our experiments revealed that the best performance on both datasets was obtained with different variants of the outlier detection method. While link prediction methods performed relatively well on single facts, their performance on compound facts was much lower than the outlier detection methods and the frequency baseline. Further analysis revealed that link prediction models are optimized to predict typical values rather than surprising ones, as they tend to regress to the centroid of the entities with similar facts.

This paper represents a first investigation of identifying surprising facts in Wikidata by using automatic methods. However, the significance of our findings is limited by the small size of the benchmarks, which cannot be expected to be representative of the size of Wikidata. We have found it challenging to create a large evaluation dataset - a key future work task is to develop a more representative evaluation set. Future work should investigate the robustness of the obtained results by evaluating on such a larger dataset, and it should develop novel surprise methods that, for example, leverage entity embeddings in ways that focus on capturing surprise. We make our code and data publicly available to facilitate subsequent work on identifying surprising facts in Wikidata: https://github.com/usc-isi-i2/surprising-facts.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-20-2-10002. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted

as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

References

[1] D. Tsurel, D. Pelleg, I. Guy, D. Shahaf, Fun facts: Automatic trivia fact extraction from wikipedia, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (2017).

[2] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85.

[3] Wikidata, Curious facts dashboard, ???? URL: https://wikidata-analytics.wmcloud.org/app/CuriousFacts.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[5] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International conference on machine learning, PMLR, 2016, pp. 2071–2080.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[7] H. Arnaout, S. Razniewski, G. Weikum, J. Z. Pan, Negative statements considered useful, Journal of Web Semantics 71 (2021) 100661.

[8] N. Klein, F. Ilievski, P. Szekely, Generating explainable abstractions for wikidata entities, in: Proceedings of the 11th on Knowledge Capture Conference, 2021, pp. 89–96.

[9] A. Prakash, M. K. Chinnakotla, D. Patel, P. Garg, Did you know?—mining interesting trivia for entities from wikipedia, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[10] I. V. Serban, A. García-Durán, C. Gulcehre, S. Ahn, S. Chandar, A. Courville, Y. Bengio, Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus, arXiv preprint arXiv:1603.06807 (2016).

[11] A. Piscopo, C. Phethean, E. Simperl, What makes a good collaborative knowledge graph: group composition and quality in wikidata, in: International Conference on Social Informatics, Springer, 2017, pp. 305–322.

[12] A. Piscopo, L.-A. Kaffee, C. Phethean, E. Simperl, Provenance information in a collaborative knowledge graph: an evaluation of wikidata external references, in: International semantic web conference, Springer, 2017, pp. 542–558.

[13] T. Safavi, D. Koutra, E. Meij, Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction, arXiv preprint arXiv:2004.01168 (2020).

[14] H. Wu, Z. Wang, X. Zhang, P. G. Omran, Z. Feng, K. Wang, A system for reasoning-based link prediction in large knowledge graphs., in: ISWC Satellites, 2019, pp. 121–124.

[15] P. Rosso, D. Yang, P. Cudré-Mauroux, Beyond triplets: hyper-relational knowledge graph embedding for link prediction, in: Proceedings of The Web Conference 2020, 2020, pp. 1885–1896.

[16] U. Joshi, J. Urbani, Searching for embeddings in a haystack: Link prediction on knowledge graphs with subgraph pruning, in: Proceedings of The Web Conference 2020, 2020, pp. 2817–2823.