That's Interesting, Tell Me More! Finding Descriptive Support Passages for Knowledge Graph Relationships

Sumit Bhatia[1], Purusharth Dwivedi[2], and Avneet Kaur[2]

[1] IBM Research AI, India
sumitbhatia@in.ibm.com
[2] IIIT Delhi, India
{purusharth14081,avneet14027}@iiitd.ac.in

Abstract. We address the problem of finding descriptive explanations of facts stored in a knowledge graph. This is important in high-risk domains such as healthcare, intelligence, etc. where users need additional information for decision making and is especially crucial for applications that rely on automatically constructed knowledge graphs where machine-learned systems extract facts from an input corpus and working of the extractors is opaque to the end-user. We follow an approach inspired from information retrieval and propose a simple, yet effective and efficient solution that takes into account passage level as well as document level properties to produce a ranked list of passages describing a given input relation. We test our approach using Wikidata as the knowledge base and Wikipedia as the source corpus and report results of user studies conducted to study the effectiveness of our proposed model.

1 Introduction

Knowledge Graphs are becoming increasingly important in knowledge and data management applications as they afford a semantic structure to the underlying data. They form crucial components of modern web search engines, state-of-the-art question answering systems such as IBM Watson, and are used in a variety of applications in domains as diverse as healthcare [28], finance [33], media [16], cybersecurity [21], etc. Entities are the fundamental units of knowledge graphs and are often presented to users as a result of a search query, or are used in applications such as exploratory search where users can search about entities of interest and browse their important relationships [19]. For various critical applications such as exploring interactions between genes and drugs [14], intelligence applications [36], etc., users may want some additional description or supporting evidence that provides some explanation of the relationship presented to them in order to build confidence in their decision making process. Even for generic information search or browsing activities the entities and relationships presented to the user may be unknown to her and thus, she may not be able to fully appreciate the relevance of information presented to her by the system. As an example, consider the relationship triple $<$*H. R. McMaster, military_rank, Lieutenant General* $>$ and its following description as extracted by our proposed approach (Section 3).

...In February 2014, Defense Secretary Chuck Hagel nominated McMaster for Lieutenant General and in July 2014, McMaster pinned on his third star when he began his duties as Deputy Commanding General of the Training and Doctrine Command and Director of TRADOCs Army Capabilities Integration Center. Army Chief of Staff General Martin Dempsey remarked in 2011 that McMaster was "probably our best Brigadier General. McMaster made Times list of the 100 most influential people in the world in April 2014...

An end-user who does not know that Mr. McMaster is a US Army officer may find the above fact much more useful when presented with the accompanying supporting description rather than presenting the fact alone. It may also help build his trust and confidence in the system. In fact, it has been found that in scenarios where users are dealing with uncertain information, use of natural language descriptions helps in the decision making process [15]. In web search engines, usefulness of small text snippets to improve end-user experience is well studied [10]. Likewise, in context of scientific digital libraries, it has been found that accompanying figures, tables, etc. with small textual descriptions helps users in judging their importance [5, 35]. Therefore, we posit that providing users with small textual explanations of the relationships may help their understanding, build their confidence in the system and help them in accomplishing their intended tasks. We believe that such a capability is even more crucial for systems that rely on knowledge graphs that are constructed automatically [29], especially using deep neural networks [37] where interpretability is a big issue.

In this work, we describe *a probabilistic method based on language models to extract supporting passages from an underlying text corpus that provide descriptive explanations of a knowledge graph relationship* (Section 3). Given an input relationship, our model takes into account passage-level and document-level evidence to rank different passages in the order of their relevance to the input relationship. Previous works on explainability of knowledge graph data have mainly focused on explaining how two entities in a graph may be related and the *explanations* are often in the form of a set of common entities or paths connecting the two entities [1, 12, 31] and thus, suffer from the same issues as discussed above. Efforts on generating textual descriptions of relationships have also focused mainly on template based methods where given a set of facts and an underlying text corpus, different templates are learned that could be used for representing the relationship [2, 43]. For example, for relations of type $< X, dateOfBirth, Y >$, sentence templates such as "X was born on Y" are learned. However, such sentences offer textual *representations* of the input relationship rather than a *supporting explanation* which is the main focus of our work. We propose an approach that is simple, effective, and unsupervised, and thus, can be easily adopted by different systems. We implemented and evaluated our approach using Wikipedia as our background text corpus and Wikidata as our knowledge base and results obtained through user studies conducted to study the effectiveness of our proposed techniques are encouraging (Section 4). The query and result sets generated by this work are also being made available for the community. We also discuss the strengths and limitations of our proposed approach and lay down directions for future work (Section 5).

2 Related Work

We provide a brief overview of related work categorized under two broad categories. First we provide an overview of most relevant papers that have looked at generating small textual descriptions of results in different search scenarios such as web search, academic search, etc. Next, we focus on works that have addressed the problem of explaining relatedness between knowledge graph entities through both graph-based and textual summaries.

2.1 Supporting Search Results With Textual Descriptions

User studies conducted by Tombros and Sanderson [41] have shown that in document retrieval systems, presenting users with short textual summaries describing the retrieved documents help them judge the importance and utility of the results much better and faster. Likewise, in Web Search Engines, it is a common practice to present results along with a small textual summary or *snippet* extracted from the web page [42] and the positive influence of snippets on end-user experience and behavior is well studied [10]. Metzger et al. [26] proposed a semantic aware document-retrieval method that transforms a given keyword query into RDF statements, and ranks documents based on their relevance to the statements. Further, the sentences matching RDF statements in the documents are extracted and presented as snippets to the user [11]. In context of academic search engines such as CiteSeer and Google Scholar, Bhatia and Mitra [6] studied the problem of generating small descriptions of *document-elements* (figures, tables, and pseudo-codes) present in academic papers to help users quickly decide their importance without having to read the whole paper. Similarly, snippets have been found useful for XML search systems [20] and ontology search systems [30] where small textual descriptions have helped users select the most suitable results for their information needs.

2.2 Explaining Knowledge Graph Relationships

Graph-Based Approaches: On receiving an entity query, Web search engines such as Google, Bing, etc. often show a list of related entities on the search page or in a separate entity box populated by information derived from the underlying knowledge base. However, it is not always apparent to the users how the suggested entities are connected to the input entity. Fang et al. [12] describe their system *REX* that takes as input two knowledge graph entities and produces a ranked list of relationships between the two entities efficiently. Bhatia et al. [3] proposed a relationship ranking function that takes into account features such as entity popularity, affinity between the input entities and strength of different relationships between them. Pirrò [31] considered the problem of explaining how two entities in a knowledge graph might be related as a sub-graph finding problem where the sub-graph consists of nodes and edges in the set of paths between the two input entities. Thus, the explanation of the relatedness between two entities is provided by means of shared entities and relationships between them. Aggarwal et al. [1] considered the task of explaining relationships between two entities as a path-ranking problem and propose a scoring mechanism to identify informative and discriminative paths.

Text Based Approaches: In context of web search where the systems present entities as part of search results, Blanco and Zaragoza [8] studied the problem of finding support sentences for explaining why an output entity is considered relevant to the original ad-hoc text query by the user. Saldanha et al. [34] addressed the problem of generating descriptions of lesser known companies and describe a template based approach to create such descriptions by generating sentences from RDF triples found in DBPedia and Freebase about the company. These sentences are generated by utilizing the RDF triples and corresponding Wikipedia sentences for known companies and learning templates such as "$< company >$ was founded by $< founder >$". Voskarides et al. [44] describe a learning to rank based sentence extraction and ranking method to find human readable descriptions of a relationship between two knowledge graph entities. Their follow-up work [43] tackles the problem using a template based approach. For a given relationship type, they identify representative sentences describing some of the relationship instances and then generating textual description of other instances of the same relationship type by selecting a suitable template and filling it with appropriate entities. Such template based approaches requires manual construction of templates for each relationship type that may be difficult for many practical applications. For example, Wikidata contains more than 1600 unique relationships types, DBPedia contains more than 2800 relationship types. The problem is exacerbated in domain specific knowledge graphs where domain knowledge is required for generating appropriate templates. Further, machine learning of such templates or other learning based methods require significant amount of training data and it may not always be feasible due to lack of such data and thus, may only be useful for a few specific relationship types.

3 Proposed Approach

Let us consider a relationship $\mathcal{R} =< s, r, t >$ in a knowledge Graph $\mathcal{K}$ where s and t correspond to the source and target nodes (entities), respectively, and r is the relationship edge label. Let P be the set of passages extracted from an underlying text corpus[3]. We wish to rank the passage $p \in P$ based on the probability that it contains a descriptive explanation of $\mathcal{R}$. Mathematically, having observed the relationship $\mathcal{R}$, we are interested in computing the probability that passage p is relevant to $\mathcal{R}$, i.e., $P(p|\mathcal{R})$. By application of Bayes' Theorem, we have:

$$P(p|\mathcal{R}) = \frac{P(p) \times P(\mathcal{R}|p)}{P(\mathcal{R})} \propto P(p) \times P(\mathcal{R}|p) \tag{1}$$

Here, $P(\mathcal{R})$ in the denominator has been ignored as it will be same for all the passages $p \in P$. The component $P(p)$ can be interpreted as the prior probability of the passage p being of interest. Note that this prior is independent of the relationship (query) and can be used to model certain domain specific characteristics based on the application requirements. For example, in a medical domain application, passages coming from

[3] Given a text corpus, there are multiple ways of extracting passages and the approach for ranking these passages is independent of the way passages are extracted. We detail our choice of passage extraction method in the section on experiments (Section 4).

a peer-reviewed article can be assigned a higher prior than passages coming from a non-authoritative article. In this work, we are focused on the general performance of the framework and hence, we assume a uniform prior as is common in information retrieval [25, Chapter 12] and thus, $P(p)$ can also be ignored for ranking purposes. With these assumptions and assuming conditional independence of three components of the relationship $\mathcal{R}$ (namely, s, r, and t), equation 1 reduces as follows.

$$P(p|\mathcal{R}) \propto \underbrace{P(s|p) \times P(t|p)}_{\substack{\text{entity} \\ \text{probability}}} \times \underbrace{P(r|p)}_{\substack{\text{relationship} \\ \text{probability}}} \tag{2}$$

Here, $P(s|p)$ and $P(t|p)$ represent the probability of observing mentions of source and target entities, s and t, respectively in the passage p. Likewise, $P(r|p)$ represents the probability that relation label r is being described in passage p. In order to compute these probabilities, we adapt the query likelihood model based on multinomial unigram language model [25] that computes probability of generating a query given a text document. We can treat each passage in P as our source document and compute the probabilities of generating the entities s, t and relation r as specified in equation 2. Note that the names of entities s and t and relationship label r consist of multiple individual words and assuming conditional independence of terms, we can simplify equation 2 as follows.

$$P(p|\mathcal{R}) \propto \prod_{w \in S \cup T \cup R} P(w|p), \tag{3}$$

Here, S, and T are the sets of terms in names of source entity s and target entity t, respectively, and R is the set of terms representing the relationship r. Note that relationship labels in knowledge graphs are often created like variable names (*bornOn, citizen_of*, etc.) that are generally not used in standard written vocabulary. Further, a given relationship may be described by different synonymous terms (occupation, profession, etc.). Therefore, to account for these variations, R can be constructed by using a set of synonyms representing a given relationship type. In this work, we have chosen relationship label aliases provided by Wikidata to obtain a set of terms that could be used for representing a given relationship type. For example, for the label *date of birth*, the list of aliases as provided by Wikidata[4] includes *born on, birthday, DOB,* etc. We note that depending upon the application at hand, different domain specific synonyms can also be used for this purpose.

Another important consideration is that a typical passage is only a few sentences long. As a result, a given passage alone may not have sufficient information to reliably approximate the probability of observing a term from the passage due to data sparsity issues. The probabilities are over estimated for the terms that are present in the passage and are under estimated for the terms that are not present in the passage. This is especially exacerbated in case of entity names (nouns) that are often mentioned as corresponding pronouns (his, her, she, etc.). As a result, a highly useful passage may get a very low score

[4] Details of *date of birth* relationship label (also called as property in Wikidata): https://www.wikidata.org/wiki/Property:P569

if the entity of interest is mentioned by its pronoun in the passage. Likewise, it is possible that a non-relevant passage may get a very high score because of multiple occurrences of just one or two terms in the passage. In order to account for such imbalances, the probability estimations are smoothed by adding document and collection level statistics. Consequently, the unigram language model of passage p is then modeled as a mixture of passage, document, and collection (corpus) language models, respectively, as follows:

$$P(w|p) = P(w|\Theta_{MM}) \tag{4}$$

$$= \lambda_1 \underbrace{P(w|\Theta_p)}_{\substack{\text{passage-level} \\ \text{evidence}}} + \lambda_2 \underbrace{P(w|\Theta_d)}_{\substack{\text{document-level} \\ \text{evidence}}} + \lambda_3 \underbrace{P(w|\Theta_c)}_{\substack{\text{collection-level} \\ \text{evidence}}} \tag{5}$$

where, $\lambda_1 + \lambda_2 + \lambda_3 = 1$. We set $\lambda_1 = 0.6, \lambda_2 = \lambda_3 = 0.2$ for our experiments. The values are chosen to give relatively more weight to passage level evidence and use document and collection level evidence as normalizing factors.

Modeling the entity probabilities and smoothing as just described serves multiple objectives. First, it helps overcome the sparsity problem due to the short length of the passage. Second, the document level evidence gives a higher score to passages that come from documents that talk more about the entities involved in input relationship. Thus, passages coming from documents that are majorly about the involved entities are given a higher weight by the ranking function described in equation 5. Also note that such a formulation also addresses the problem of co-reference resolution [17] to some extent and can be interpreted as a probabilistic variant of the heuristic used by Wu and Weld [46] that replaces most frequent pronouns in Wikipedia article with article title. Lastly, the collection level evidence is also important as it plays the role of a reference or *background* language model and provides term weighing similar to inverse document frequency (IDF) [48].

The individual probabilities in equation 5 can be computed by using the statistics from passage, document, and collection as follows:

$$\text{Passage Evidence: } P(w|\theta_p) = \frac{count(w,p)+1}{|p|+|V|} \tag{6}$$

$$\text{Document Evidence: } P(w|\theta_d) = \frac{count(w,d)+1}{|d|+|V|} \tag{7}$$

$$\text{Collection Evidence: } P(w|\theta_c) = \frac{count(w,c)}{|C|} \tag{8}$$

Here, V is the vocabulary of the corpus and $|\cdot|$ indicates the size of the set. Note that we have added the constant one in equations 6 and 7 to prevent zero probabilities for terms that may not be present in the respective passage or document. Further, the denominators are chosen so that the sum of probabilities over the entire vocabulary is one. Also note that the additive factor is not required in the collection model as all the terms in the vocabulary are present in the collection by definition.

4 Experimental Evaluation

4.1 Data Description

In this section we discuss the dataset used in our experiments and how the queries and relevance judgments were obtained. The resulting resources (queries, results, and relevance judgments, and parameters used) are being made available to the community through our git repository[5].

Relationship Queries: We need relationship triples of the form $< s, r, t >$ that will constitute our query relationships for which the supporting passages need to be retrieved from the underlying corpus. In order to create such a query set, we selected titles of the top 25 most viewed pages[6] each for the months of January-April, 2017. From these 100 (25 for each month) page titles, we retained only those that correspond to named entities by manually filtering out titles like *List of Black Mirror episodes, Deaths in 2017*, etc. That gave us a total of 80 unique entities. Next, we used Wikidata[7] as our knowledge base and retrieved all relationships of the entities selected previously using the SPARQL end-points provided by Wikidata. From all these retrieved relationships, we manually filtered out the relationships that were not in English language, were of type *instance of* and *subclass of*, and, where the target entity was not a named entity. This resulted in a final set of 1250 unique relationship triples from which we selected 150 triples at random as our final relationship query set that was used in subsequent experiments.

Source Corpus and Passages: We chose Wikipedia[8] as our underlying corpus. There are multiple ways to extract a set of passages given a text corpus such as utilizing the document structure and paragraph or section markers present in the documents itself. However, the passages thus extracted are usually very long, often running into tens of sentences. Further, while such paragraph or section markers are available for well-structured corpora such as Wikipedia, they may not always be available for different source documents. More importantly, such long passages may be detrimental to the end-user experience as they consume valuable screen real estate and reading them requires significant additional efforts from users. Another option is to use text segmentation methods such as TextTiling [18] that segment the input text into topically coherent passages. However, such approaches require significant pre-processing efforts, especially for large corpora (few millions of documents) often encountered in real world applications. In practice, simple (and *fast*) segmentation of input text into fixed length, overlapping passages using a sliding window approach is found to be equally effective [4, 22, 39, 40], if not better, and is the approach we also take. Use of overlapping passages is also encouraged as it reduces the chances of relevant information getting split between two consecutive passages [9]. Therefore, we split the input text of each document into overlapping passages of three consecutive sentences using a sliding

[5] https://github.com/sumit-research/kg-support-passages
[6] https://en.wikipedia.org/wiki/Wikipedia:Top_25_Report
[7] https://www.wikidata.org/wiki/Wikidata:Main_Page
[8] Specifically, we used the dump of 20^{th} April, 2017.

window of size three as suggested by Spangler et al. [38]. This resulted in about 80.5 million extracted passages that constitute our source set of passages (set P in Section 3). Note that one drawback of such a pre-processing is that multiple overlapping passages containing a highly relevant sentence can all appear in top positions in the final ranked list, thereby artificially boosting the proportion of relevant passages and at the same time, causing a degraded user experience due to repetitive results. Therefore, we perform a post-processing step where such repetitions are detected and only the highest scoring passage is retained and the rest of the overlapping passages are removed from the final ranked list.

In order to compute the different passage, document, and collection based statistics, we used the Indri toolkit provided by the Lemur project[9]. The toolkit offers capabilities to query and index a collection of documents, and APIs to compute term statistics required for language model based computations described in our ranking function (equation 5). Specifically, we created two indexes using Indri – a *passage index* of all the extracted passages to compute passage level statistics and an *article index* of all the Wikipedia articles (about 5.34 million articles) to compute document and collection level statistics. A standard stopword list provided by the Onix text retrieval toolkit[10] was used to filter out common stop words and stemming was performed using Porter's Stemmer [32]. The parameter files used for creating and querying the indexes can be found in our git repository.

Baseline Methods: We use the inference network based generative passage retrieval algorithm implemented in Indri [27] as our first baseline method (*Inf. N/w*). This is a state-of-the-art passage retrieval method and is often chosen as a baseline for various research tasks related to passage retrieval [45, 47]. Given a query, this method finds documents that are relevant for the query and then extracts specific continuous portions of text from the documents that are highly relevant for the query. Given an input relationship tuple $< s, r, t >$, the input query to Indri consists of all the terms in source and target entity names and relationship description. Next, as our second baseline method, we add query expansion [25, Chapter 9] on top of the first baseline by using relationship aliases (as described in Section 3). We denote this baseline as *Inf. N/w+Rel.Exp* in subsequent discussions. Note that for passage retrieval, Indri requires passage length as an input parameter. For comparison purposes, we specify the length of passages to be returned by Indri as 600 characters as this is the average length of passages extracted by our proposed approach. Further, note that due to fixed length of the passages retrieved, it is possible that the retrieved passaged are often truncated and thus, have incomplete sentences. In order to overcome this shortcoming, such truncated sentences are completed in a post-processing step so that the extracted passages are well-formed.

4.2 Effectiveness Evaluation

In order to study the effectiveness of our proposed approach for finding high quality descriptive passages, we selected a random set of 50 relationship triples from the set

[9] https://www.lemurproject.org/
[10] http://www.lextek.com/manuals/onix/stopwords1.html

	Evaluator 1	Evaluator 2	Evaluator 3	Final
non-relevant	406	438	444	449
partially-relevant	41	11	43	12
relevant	248	246	208	234

Table 1. Distribution of the labels assigned by the three evaluators. There were a total of 695 $< query, passage >$ pairs and each evaluator provided judgments for all 695 pairs. Last column reports the results after combining all the judgments where the final rating of a $< query, passage >$ pair was decided after taking the majority vote.

of 150 triples described above. For each of these 50 triples, we retrieved the top five passages from the corpus ranked by our ranking function (equation 5). We also obtained five passages for each relationship triple by the two baseline methods. This resulted in 695 unique $< query, passage >$ pairs. Note that this number is less than 750 (50 queries $\times$ 3 methods $\times$ 5 passages) because in some cases, the same passage was retrieved by multiple methods.

Next, we took the help of three human evaluators to evaluate the quality and correctness of the passages retrieved by the baselines and our proposed method. The evaluators were advanced graduate students in Computer Science, not associated with the project, and had good command of the English language. The evaluators worked independently of each other and were compensated monetarily for their efforts.

For each of the 50 queries used in this study, all the extracted passages for that query by the three methods were presented to the evaluators in a randomized order and they were not informed which passage was retrieved by which method. The evaluators were asked to rate each passage on three point scale – 0 if the passage is incorrect, irrelevant or not at all useful; 1 if the passage contains the relationship but is only partially relevant and does not provide a good explanation; and 2 if the passage is correct and highly relevant and provides a good explanation. All three evaluators provided their judgments for all the 695 $< query, passage >$ pairs.

Inter Annotator Agreement: We used Fleiss' Kappa coefficient [13] to measure the agreement between the three evaluators. The value of Kappa coefficient was computed to be 0.67, indicating substantial agreement. For 695 $< query, passage >$ pairs, all three evaluators agreed on the label 545 times, two evaluators provided the same label 130 times and for 20 pairs, all three evaluators provided different ratings. In case of conflict, the final label for a $< query, passage >$ pair was decided by the majority vote and 20 pairs where all three evaluators disagreed were assigned a label of 0 (irrelevant). Table 1 provides further details about the distribution of evaluations provided by the three evaluators.

Results: Table 2 compares the three approaches by using precision, precision at rank 1 ($P@1$), and mean reciprocal rank (MRR). While precision measures how many of the passages extracted by each method are relevant, $P@1$ and MRR measure the ability of

the respective methods to identify a relevant passage as the top-ranked passage. This is important because in real world applications, due to limited screen real estate and to minimize users' efforts, we want to present the best results at the top position. As can be observed, the proposed approach achieves a $P@1$ of 0.86 compared to 0.251 and 0.165 for the baseline methods. Similar out-performance is observed in the case of MRR values. Further, we note that the proposed approach achieves an overall precision of 0.727 compared to 0.156 and 0.088 for the baselines. Next, for a fine-grained analysis, Table 3 provides the distribution of passages marked as irrelevant, partially relevant, and highly relevant for the three approaches. We note that for the proposed approach, only about 16% of the passages were found to be irrelevant by the evaluators compared to about 80% for the baseline approaches. These results indicate not only that the proposed approach is able to retrieve a lot more relevant passages describing the input query relationship (as indicated by precision), it is also able to offer relevant results at top positions (as indicated by $P@1$ and MRR values).

A surprising observation from these results is the poor performance of the *Inf. N/w+Rel. Exp.* baseline method, even when compared with the plain *Inf. N/w* method. Aliases of relationship labels were incorporated in order to enable the *Inf. N/w* method to identify passages where variations of relationship terms are used. However, on analysis of the retrieved passages, we observed that addition of the alias terms led to retrieval of many passages that talked about the relationship label in general. For example, for the query $< MariahCarey, spouse, NickCannon >$, the following passage is retrieved that talks about the concept of *spouse* in general.

> *Wife: Intro A wife is a female partner in a continuing marital relationship. A wife may also be referred to as a spouse, which is a gender-neutral term. The term continues to be applied to a woman who has separated from her partner, and ceases to be applied to such a woman only when her marriage has come to an end, following a legally recognized divorce or the death of her spouse. On the death of her partner, a wife is referred to as a widow, but not after she is divorced from her partner.*

Evidently, the above passage contains multiple mentions of different aliases of the *spouse of* relationship[11] and thus, this passage got a very high score. This example illustrates the strength of the proposed approach that avoids such a dominance of certain terms in the passage by incorporating the document and collection level evidences in the ranking function (Eq. 5) that assigns lower score to passages from documents that contain little or no information about the entities involved in the relationship.

4.3 Preference Evaluation

In this section, we describe the experiment conducted to study the preferences of end-users when passages extracted by different approaches are presented to them side by side. We chose only the *Inf. N.w* baseline for comparison with the proposed approach

[11] Aliases of spouse include husband, wife, married to, consort, partner, marry, marriage, partner, married, wedded to, wed, and life partner. https://www.wikidata.org/wiki/Property:P26

	P@1	Precision	MRR
Inf. N/w	0.251	0.156	0.272
Inf. N/w + Rel. Exp.	0.165	0.088	0.144
Proposed Approach	**0.860**	**0.727**	**0.805**

Table 2. Performance of the baseline methods and proposed approach as measured by precision, P@1, and MRR.

	No. of passages marked as		
	irrelevant	**partially-relevant**	**highly relevant**
Inf. N/w	198	4	43
Inf. N/w + Rel. Exp.	210	3	32
Proposed Method	41	5	159

Table 3. Distribution of judgment labels for the baseline and proposed approach. Note that the total number of passages for proposed approach is 246 instead of 250 because some passages appeared for more than one query.

due to its superior performance compared with the other baseline method. For this experiment, we used the full set of 150 relationship tuples (Section 4.1) and recruited 3 undergraduate computer science students that were not associated with this project and were compensated monetarily for their efforts. For each query, the top scored passages extracted by the baseline and our proposed approach were presented to the evaluators side by side and they were asked to chose from one of the following four options: *(i)* both passages are equally good/useful, *(ii)* both passages are equally bad, *(iii)* passage on the left offer a better description, and *(iv)* passage on the right offers a better description. Note that the order in which the passages were presented to the evaluators was randomized and they were not informed of the method that produced a specific passage. Each evaluator provided preference judgments for 50 relationships. The results are summarized in Table 4. As can be seen from the results, for an overwhelming majority of the time, all the evaluators preferred the passages extracted by the proposed approach. Overall, more than 50% of the times, passages retrieved by the proposed approach were preferred (73 out of 150) whereas the passages retrieved by the baseline method was preferred only 11 times.

5 Discussions

In this section, we provide some representative examples to illustrate the strengths and weaknesses of our proposed approach and discuss possible future directions of research. Consider the relationship <*John Cena, nickname, The Protoype*>, for which the passages as produced by the baseline and our proposed approach are as follows.

> ***Baseline:*** *A prototype is something that is representative of a category of things, or an early engineering version of something to be tested. Prototype may also*

	Both Not Useful	Both Equally Useful	Baseline	Proposed
Evaluator 1	14	13	3	20
Evaluator 2	6	17	2	25
Evaluator 3	10	6	6	28
Total	30	36	11	73

Table 4. Comparative evaluations provided by the three evaluators when presented with top passages from the baseline and proposed approach side by side. Each evaluator provided evaluations for 50 relationship queries.

> *refer to: Automobiles. Citroën Prototype C, a range of vehicles created by Citroën from 1955 to 1956 Citroën Prototype Y, a project of replacement of the Citroën Ami studied by Citroën in the early seventies Daytona Prototype, a sports ca*
> ***Proposed approach:*** *In 2001, Cena signed a developmental contract with the WWF and was assigned to its developmental territory Ohio Valley Wrestling (OVW). During his time there, Cena wrestled under the ring name The Prototype and held the OVW Heavyweight Championship for three months and the OVW Southern Tag Team Championship (with Rico Constantino) for two months. Throughout 2001, Cena would receive four tryouts for the WWF main roster, as he wrestled multiple enhancement talent wrestlers on both WWF house shows and in dark matches before WWF television events.*

Note that the first passage contains multiple occurrences of the word *prototype* which is also a less frequent word in the corpus, and thus was highly ranked by the baseline approach. On the other hand, the passage produced by the proposed approach is able to correctly identify a good passage even though it only had one occurrence of *prototype*. One reason for this passage getting a very high score is the document level component of the ranking function (equation 5). This passage comes from the Wikipedia article about *John Cena* and thus, its score was boosted by the document-evidence component.

Error Analysis: By further analyzing the passages extracted by the proposed approach and feedback from the evaluators, we observed two major characteristics of the passages that were not rated as relevant by the evaluators. In the first category, while the extracted passage does talk about the entities involved, it does not provide any description of the relationship specified in the query. Consider the following passage for the relationship *<Alan Comes, employer, Fox News>*.

> *...Goldlines television advertising includes cable networks such as CNN, CNBC, Fox News, History International and Fox Business. Goldline has also been the sponsor of the shows of a number of conservative radio and television hosts, including The American Advisor, and The Glenn Beck Program, The Laura Ingraham Show, The Fred Thompson Show, The Huckabee Report, The Lars Larson Show, The Monica Crowley Show, The Mark Levin Show, and The Alan*

Colmes Show. In 2009, Goldline incorrectly labeled Glenn Beck as a paid spokesman on its website which raised concerns with his employer, Fox News, which prohibit such a relationship; they later corrected it to radio sponsor...

This passage got a high score by the proposed scoring function because it talks about Fox News and Alan Comes and the originating document also has other mentions of Fox News. However, it does not provide any description about the employment of Alan Comes at Fox News. Instead, it provides a lot of unnecessary information to the user.

The other type of passages that were not judged relevant by the evaluators were the ones that made an indirect reference to the relationship query. Consider the following passage for the query $<$*Warren Beatty,occupation,Film Producer*$>$.

In 1994, Astin directed and co-produced (with his wife, Christine Astin) the short film Kangaroo Court, which received an Academy Award nomination for Best Live Action Short Film. Astin continued to appear in films throughout the 1990s, including the Showtime science fiction film Harrison Bergeron (1995), the Gulf War film Courage Under Fire (1996), and the Warren Beatty political satire Bulworth (1998). After The Goonies, Astin appeared in several more films, including the Disney made-for-TV movie, The B. R. A

Here again, the passage contains a lot of unnecessary information and only contains a fleeting reference to Warren Beatty and the movie Bulworth. There is no explicit mention here that Warren Beatty is a film producer and thus, the evaluators did not find this passage to be very informative.

Directions for Future Work: In the present work, we focused on relationship triples between two named entities. It will be interesting to extend the proposed models to triples where the target is a data value instead of a named entity (e.g. $<$ *Burj Khalifa, height, 828 metres* $>$). This is a challenging problem as the information in text could be present in multiple formats (numbers, text, etc.) as well as in different units. Another aspect of our proposed approach that merits further research is handling of negations. For example, consider the relationship $< X, spouseOf, Y >$ and a sentence, *X is not wife of Y*. Such a sentence will also be considered a relevant sentence by our method even though it offers negative evidence of the fact under consideration. However, handling negations in text is a hard problem and is an active area of research [23]. One related interesting application of our proposed approach that is worth exploring further is in *fact checking* systems such as DeFacto [24] where the users could query for supporting evidence for facts presented to them and can evaluate if the information shown to them is correct.

Another direction for future work is to combine the proposed approach with existing methods for entity search and recommendation [7] and path ranking [1, 12, 31], and offer textual descriptions for how two entities in the knowledge graph may be related. Such techniques will be useful for discovery and exploratory search based applications and may improve end-user experience by offering human readable explanations of systems' graphical output.

6 Conclusions

We studied the problem of providing descriptive explanations for relationships in a knowledge graph and described a probabilistic method for ranking passages derived from an input corpus in order of their relevance to the input relationship. The proposed method is simple, effective, and outperformed state-of-the-art baseline methods in user studies conducted for evaluating the effectiveness of our proposed approach. We presented some representative examples to illustrate the strengths and weaknesses of our approach and provided directions for future work.

References

1. Aggarwal, N., Bhatia, S., Misra, V.: Connecting the dots: Explaining relationships between unconnected entities in a knowledge graph. In: The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers. pp. 35–39 (2016)
2. Althoff, T., Dong, X.L., Murphy, K., Alai, S., Dang, V., Zhang, W.: Timemachine: Timeline generation for knowledge-base entities. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 19–28. ACM (2015)
3. Bhatia, S., Goel, A., Bowen, E., Jain, A.: Separating wheat from the chaff – a relationship ranking algorithm. In: The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers. pp. 79–83 (2016)
4. Bhatia, S., He, B., He, Q., Spangler, S.: A scalable approach for performing proximal search for verbose patent search queries. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 2603–2606. ACM (2012)
5. Bhatia, S., Lahiri, S., Mitra, P.: Generating synopses for document-element search. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. pp. 2003–2006. CIKM '09, ACM, New York, NY, USA (2009)
6. Bhatia, S., Mitra, P.: Summarizing figures, tables, and algorithms in scientific publications to augment search results. ACM Trans. Inf. Syst. 30(1), 3:1–3:24 (Mar 2012)
7. Bhatia, S., Vishwakarma, H.: Know Thy Neighbors, and More! Studying the Role of Context in Entity Recommendation. In: Proceedings of the 29th ACM Conference on Hypertext and Social Media. HT '18, ACM, New York, NY, USA (2018)
8. Blanco, R., Zaragoza, H.: Finding support sentences for entities. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 339–346. ACM (2010)
9. Callan, J.P.: Passage-level evidence in document retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 302–310. Springer-Verlag New York, Inc. (1994)
10. Clarke, C.L., Agichtein, E., Dumais, S., White, R.W.: The influence of caption features on clickthrough patterns in web search. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 135–142. ACM (2007)
11. Elbassuoni, S., Hose, K., Metzger, S., Schenkel, R.: Roxxi: Reviving witness documents to explore extracted information. Proceedings of the VLDB Endowment 3(1-2), 1589–1592 (2010)
12. Fang, L., Sarma, A.D., Yu, C., Bohannon, P.: Rex: explaining relationships between entity pairs. Proceedings of the VLDB Endowment 5(3), 241–252 (2011)

13. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychological bulletin 76(5), 378 (1971)
14. Fokoue, A., Sadoghi, M., Hassanzadeh, O., Zhang, P.: Predicting drug-drug interactions through large-scale similarity-based link prediction. In: Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains - Volume 9678. pp. 774–789. Springer-Verlag New York, Inc., New York, NY, USA (2016)
15. Gkatzia, D., Lemon, O., Rieser, V.: Natural language generation enhances human decision-making with uncertain information. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers (2016)
16. Gutiérrez-Cuellar, J., Gómez-Pérez, J.M.: Havas 18 labs: A knowledge graph for innovation in the media industry. In: Polleres, A., Castro, A.G., Benjamins, R. (eds.) International Semantic Web Conference (Industry Track). CEUR Workshop Proceedings, vol. 1383 (2014)
17. Hajishirzi, H., Zilles, L., Weld, D.S., Zettlemoyer, L.: Joint coreference resolution and named-entity linking with multi-pass sieves. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 289–299 (2013)
18. Hearst, M.A.: Texttiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics 23(1), 33–64 (1997)
19. Heim, P., Lohmann, S., Stegemann, T.: Interactive relationship discovery via the semantic web. In: The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 –June 3, 2010, Proceedings, Part I. pp. 303–317 (2010)
20. Huang, Y., Liu, Z., Chen, Y.: Query biased snippet generation in xml search. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 315–326. ACM (2008)
21. Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., Ferragut, E., Goodall, J.: Developing an ontology for cyber security knowledge graphs. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference. pp. 12:1–12:4. CISR '15, ACM, New York, NY, USA (2015)
22. Khalid, M.A., Verberne, S.: Passage retrieval for question answering using sliding windows. In: Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering. pp. 26–33. Association for Computational Linguistics (2008)
23. Konstantinova, N., De Sousa, S.C., Díaz, N.P.C., López, M.J.M., Taboada, M., Mitkov, R.: A review corpus annotated for negation, speculation and their scope. In: Lrec. pp. 3190–3195 (2012)
24. Lehmann, J., Gerber, D., Morsey, M., Ngomo, A.C.N.: Defacto-deep fact validation. In: International Semantic Web Conference. pp. 312–327. Springer (2012)
25. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
26. Metzger, S., Elbassuoni, S., Hose, K., Schenkel, R.: S3k: seeking statement-supporting top-k witnesses. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 37–46. ACM (2011)
27. Metzler, D., Croft, W.: Combining the language model and inference network approaches to retrieval. Information Processing & Management 40(5), 735 – 750 (2004)
28. Nagarajan, M., et al.: Predicting future scientific discoveries based on a networked analysis of the past literature. In: KDD. pp. 2019–2028. KDD '15 (2015)
29. Niu, F., Zhang, C., Ré, C., Shavlik, J.W.: Deepdive: Web-scale knowledge-base construction using statistical learning and inference. VLDS 12, 25–28 (2012)
30. Penin, T., Wang, H., Tran, T., Yu, Y.: Snippet generation for semantic web search engines. In: The Semantic Web: 3rd Asian Semantic Web Conference, ASWC 2008, Bangkok, Thailand, December 8-11, 2008. Proceedings. pp. 493–507 (2008)

31. Pirrò, G.: Explaining and suggesting relatedness in knowledge graphs. In: The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I. pp. 622–639 (2015)
32. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
33. Ruan, T., Xue, L., Wang, H., Hu, F., Zhao, L., Ding, J.: Building and exploring an enterprise knowledge graph for investment analysis. In: International Semantic Web Conference. pp. 418–436. Springer (2016)
34. Saldanha, G., Biran, O., McKeown, K., Gliozzo, A.: An entity-focused approach to generating company descriptions. In: The 54th Annual Meeting of the Association for Computational Linguistics. p. 243 (2016)
35. Sandusky, R.J., Tenopir, C.: Finding and using journal-article components: Impacts of disaggregation on teaching and research practice. Journal of the Association for Information Science and Technology 59(6), 970–982 (2008)
36. Sheth, A., Aleman-Meza, B., Arpinar, I.B., Bertram, C., et al.: Semantic association identification and knowledge discovery for national security applications. Journal of Database Management 16(1), 33 (2005)
37. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. pp. 926–934 (2013)
38. Spangler, S., Kreulen, J.T., Lessler, J.: Generating and browsing multiple taxonomies over a document collection. Journal of Management Information Systems 19(4), 191–212 (2003)
39. Tiedemann, J.: Comparing document segmentation strategies for passage retrieval in question answering. In: Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'07). vol. 1 (2007)
40. Tiedemann, J., Mur, J.: Simple is best: experiments with different document segmentation strategies for passage retrieval. In: Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering. pp. 17–25. Association for Computational Linguistics (2008)
41. Tombros, A., Sanderson, M.: Advantages of query biased summaries in information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 2–10. ACM (1998)
42. Turpin, A., Tsegay, Y., Hawking, D., Williams, H.E.: Fast generation of result snippets in web search. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 127–134. ACM (2007)
43. Voskarides, N., Meij, E., de Rijke, M.: Generating descriptions of entity relationships. In: Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017. pp. 317–330 (2017)
44. Voskarides, N., Meij, E., Tsagkias, M., de Rijke, M., Weerkamp, W.: Learning to explain entity relationships in knowledge graphs. In: ACL (1). pp. 564–574 (2015)
45. Wang, M., Si, L.: Discriminative probabilistic models for passage based retrieval. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 419–426. ACM (2008)
46. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 118–127. ACL '10 (2010)
47. Yang, H., Callan, J., Si, L.: Knowledge transfer and opinion detection in the trec 2006 blog track. In: TREC (2006)
48. Zhai, C., Lafferty, J.: The dual role of smoothing in the language modeling approach. In: Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001 (2001)