An Entity-centric Approach for Overcoming Knowledge Graph Sparsity

Manjunath Hegde
Indian Institute of Science
manjunath@ssl.serc.iisc.in

Partha Talukdar
Indian Institute of Science
ppt@serc.iisc.in

Abstract

Automatic construction of knowledge graphs (KGs) from unstructured text has received considerable attention in recent research, resulting in the construction of several KGs with millions of entities (nodes) and facts (edges) among them. Unfortunately, such KGs tend to be severely sparse in terms of number of facts known for a *given* entity, i.e., have low *knowledge density*. For example, the NELL KG consists of only 1.34 facts per entity. Unfortunately, such low knowledge density makes it challenging to use such KGs in real-world applications. In contrast to *best-effort* extraction paradigms followed in the construction of such KGs, in this paper we argue in favor of ENTIty Centric Expansion (ENTICE), an *entity-centric* KG population framework, to alleviate the low knowledge density problem in existing KGs. By using ENTICE, we are able to increase NELL's knowledge density by a factor of 7.7 at 75.5% accuracy. Additionally, we are also able to extend the ontology discovering new relations and entities.

1 Introduction

Over the last few years, automatic construction of knowledge graphs (KGs) from web-scale text data has received considerable attention, resulting in the construction of several large KGs such as NELL (Mitchell et al., 2015), Google's Knowledge Vault (Dong et al., 2014). These KGs consist of millions of entities and facts involving them. While measuring size of the KGs in terms of number of entities and facts is helpful, they don't readily capture the volume of knowledge needed in real-world applications. When such a KG is used in an application, one is often interested in known facts for a *given* entity, and not necessarily the overall size of the KG. In particular, knowing the average number of facts per entity is quite informative. We shall refer to this as the *knowledge density* of the KG.

	Known Target **E**ntity	**N**ew Target **E**ntity
Known **R**elation	KR-KE	KR-NE
New **R**elation	NR-KE	NR-NE

Table 1: Any new fact involving a source entity from a Knowledge Graph (i.e., facts of the form *entity1-relation-entity2* where *entity1* is already in the KG) can be classified into one of the four extraction classes shown above. Most KG population techniques tend to focus on extracting facts of the KR-KE class. ENTICE, the entity-centric approach proposed in this paper, is able to extract facts of all four classes.

Low knowledge density (or high sparsity) in automatically constructed KGs has been recognized in recent research (West et al., 2014). For example, NELL KG has a knowledge density of 1.34. Such low knowledge density puts significant limitations on the utility of these KGs. Construction of such KGs tend to follow a batch paradigm: the knowledge extraction system makes a full pass over the text corpus extracting whatever knowledge it finds, and finally aggregating all extractions into a graph. Clearly, such *best-effort* extraction paradigm has proved to be inadequate to address the low knowledge density issue mentioned above. We refer to such paradigm as best-effort since its attention is divided equally among all possible entities.

Recently, a few *entity-centric* methods have

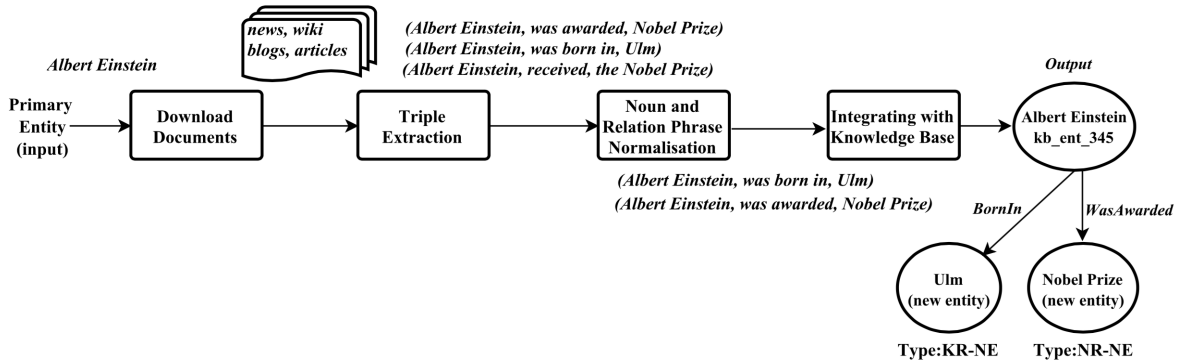

Figure 1: Dataflow and architecture and of ENTICE. See Section 3 for details.

been proposed to increase knowledge density in KGs (Gardner et al., 2013; Gardner et al., 2014). In contrast to the best-effort approaches mentioned above, these entity-centric approaches aim at increasing knowledge density for a *given* entity. A new fact involving the given entity can belong to one of the four types shown in Table 1. Unfortunately, these densifying techniques only aim at identifying instances of known relations among entities already present in the KG, i.e., they fall in the KR-KE type of Table 1.

In this paper we propose ENTIty Centric Expansion (ENTICE), an entity-centric knowledge densifying framework which, given an entity, is capable of extracting facts belonging to all the four types shown in Table 1. By using ENTICE, we are able to increase NELL's knowledge density by a factor of 7.7[1], while achieving 75.4% accuracy. Our goal here is to draw attention to the effectiveness of entity-centric approaches with bigger scope (i.e., covering all four extraction classes in Table 1) towards improving knowledge density, and that even relatively straightforward techniques can go a long way in alleviating low knowledge density in existing state-of-the-art KGs. ENTICE code is available at: `https://github.com/malllabiisc/entity-centric-kb-pop`

2 Related Work

Open Information Extraction (OIE) systems (Yates et al., 2007; Fader et al., 2011; Schmitz et al., 2012) aim at extracting textual triples of the form noun phrase-predicate-noun phrase. While such systems aim for extraction coverage, and because they operate in an ontology-free setting, they don't directly address the problem of improving knowledge density in ontological KGs such as NELL. However, OIE extractions provide a suitable starting point which is exploited by ENTICE.

(Galárraga et al., 2014) addresses the problem of normalizing (or canonicalizing) OIE extractions which can be considered as one of the components of ENTICE (see Section 3.3).

As previously mentioned, recent proposals for improving density of KGs such as those reported in (Gardner et al., 2013; Gardner et al., 2014) focus on extracting facts of one of the four extraction classes mentioned in Table 1, viz., KR-KE. The KBP challenge (Surdeanu, 2013) also focuses on extracting facts while keeping the relation set fixed, i.e., it addresses the KR-KE and KR-NE extraction classes.

A method to improve knowledge density in KGs by using search engine query logs and a question answering system is presented in (West et al., 2014). The proprietary nature of datasets and tools used in this approach limits its applicability in our setting.

ENTICE aims to improve knowledge density by extracting facts from all four extraction classes, i.e., for a given entity, it extracts facts involving known relations, identifies potentially new relations that might be relevant for this entity, establishes such relations between the given entity and other known as well as new entities – all in a single system. While various parts of this problem have been studied in isolation in the past, ENTICE is

[1]Measured with respect to the five categories experimented with in the paper. See Section 4 for details.

the first system to the best of our knowledge that addresses the complete problem as a single framework.

3 ENTIty Centric Expansion (ENTICE)

Overall architecture and dataflow within ENTICE is shown in Figure 1. We describe each of the components in the sections below.

3.1 Data Preprocessing

Given the source entity, documents relevant to it are downloaded by issues queries against Google. In order to make the query specific, especially in case of ambiguous entities, a few keywords are also added to the query. For the experiments in this paper, the category is used as the keyword. For example, for the entity *Albert Einstein* from the *scientist* category, the query will be *"Albert Einstein scientist"*. Top 20 documents returned by the search engine are downloaded and processed further. Text is extracted from the raw downloaded documents using regex patters, HTML tag matching, and by using the Boilerpipe tool[2].

3.2 Triple Extraction

Text of each document obtained in the previous step is processed through the Stanford CoreNLP toolkit (Manning et al., 2014) for tokenization, coreference resolution, and dependency parsing. Tokenized and coreference-resolved sentences are then passes through OpenIEv4 system [3] to extract *(noun phrase, predicate, noun phrase)* triples. Multiple and overlapping triples from the sentence was permitted. Length filter is applied on the noun phrase and the predicate of the triple extracted. This eliminates triples whose predicate is more than 6 tokens and noun phrase more than 7 tokens.

3.3 Noun and Relation Phrase Normalization

Noun phrases (NPs) and relation phrases obtained from the previous step are normalized (or canonicalized) in this step. Canopy clustering technique as proposed in (Galárraga et al., 2014) was used for noun phrase as well relation phrase clustering. Initial clustering is done over the *unlinked* noun phrases in the triples. Please note that since we are working in an entity-centric manner, one of the two NPs present in the triple is already connected to the knowledge graph, and hence is considered *linked*. To cluster noun phrases, we first construct canopies corresponding to each word in the noun phrase. For example, for noun phrase *Albert Einstein*, we create two canopies, viz., a canopy for *Albert* and another canopy for *Einstein*, and add *Albert Einstein* to both canopies. Grouping of noun phrases inside the canopy is the next step of clustering phase. Noun phrase similarity is calculated based on similarity of words in the noun phrases. Word similarity is either direct string matching or Gensim similarity score[4], which internally uses word2vec embeddings (Mikolov et al., 2013). After calculating pairwise similarity of noun phrases, hierarchical clustering is carried out to group noun phrases inside each canopy. A threshold score is used to stop hierarchical clustering. At the end of this process, we have canopies and groups of noun phrases inside them. A noun phrase can be in more than one canopy, hence those groups across canopies are merged if the similarity is greater than certain threshold. After this, each group will contain facts which have similar noun phrases and different (or same) relation phrase. Again the facts are clustered based on the similarity of the relation phrase. Relation phrase similarity calculation step resembles the one used for noun phrases as described above.

After this triple clustering step, the best representative triple from each cluster is selected based on a few rules. We consider the structure of POS tags in noun phrases of a triple as one of the criteria. Secondly, if both noun phrases in the triple are linked to the knowledge graph, then it makes the triple more likely to become a representative tuple of the cluster. Also, if the NPs present in the triple are frequent in the cluster, then it makes the corresponding triple more like to become a representative.

[2]Boilerpipe: http://code.google.com/p/boilerpipe
[3]OpenIEv4: http://knowitall.github.io/openie/
[4]https://github.com/piskvorky/gensim/

Category	Knowledge Density in NELL	Knowledge Density after ENTICE	# Facts Evaluated	# Correct Facts	Accuracy
Scientist	1.27	18.5	164	141	85.97
Universities	1.17	9	197	141	71.57
Books	1.34	4.49	202	165	81.68
Birds	1.27	6.69	194	136	70.10
Cars	1.5	11.61	201	140	69.65
Overall	1.3	10.05	958	723	75.46

Table 2: Knowledge densities of five categories in NELL and after application of ENTICE, along with resulting accuracy. We observe that overall, ENTICE is able to increase knowledge density by a factor of 7.7 at 75.5% accuracy. This is our main result.

Entity Name	All facts in NELL	Sample facts extracted by ENTICE	Extraction Class
George Paget Thomson	*(George Paget Thomson, isInstanceOf, scientist)*	*(Sir George Thomson, isFellowOf, Royal Society)*	NR-KE
		(George Thomson, hasSpouse, Kathleen Buchanan Smith)	KR-NE
		(George Paget Thomson, diedOn, September 10)	KR-KE

Table 3: Facts corresponding to an entity from the *scientists* domain in NELL as well as those extracted by ENTICE. While NELL contained only one fact for this entity, ENTICE was able to extract 15 facts for this entity, only 3 of which are shown above.

Category	KR - KE			KR - NE			NR - KE			NR - NE		
	correct facts	wrong facts	acc.	correct facts	wrong facts	acc.	correct facts	wrong facts	acc.	correct facts	wrong facts	acc.
Scientists	57	10	85.07	61	8	88.40	14	3	82.35	9	2	81.81
Cars	68	35	66.01	58	21	73.41	9	5	64.28	5	0	100
Universities	52	30	63.41	68	20	77.27	9	2	81.81	12	4	75
Books	78	24	76.47	79	12	86.81	2	0	100	6	1	85.71
Birds	67	29	69.79	46	19	70.76	15	4	78.94	8	6	57.14
Overall	322	128	71.55	312	80	79.59	49	14	77.77	40	13	75.47

Table 4: Accuracy breakdown over ENTICE extractions for each of the four extraction classes in Table 1. For each category, approximately 200 extractions were evaluated using Mechanical Turk.

3.4 Integrating with Knowledge Graph

The set of normalized triples from the previous step are linked with the Knowledge Graph, whenever possible, in this step. For a given normalized triple, following steps are performed as part of linking. First, category of each noun phrase in the triple is obtained based on string matching. In case of no match, refinements like dropping of adjectives, considering only noun phrases are done to for rematching. Now, the relation phrase is mapped to an existing predicate in the KG based on the extraction patterns in the metadata of the target relation (e.g., NELL and many other KGs have such metadata available). Candidate predicates are chosen from the above mapped predicates based on category signature of the two noun phrases (i.e. entity1 and entity2). This is possible since the all the predicates in NELL have the type signature defined in the metadata. Frequency of the relation phrase in the metadata is used as a criteria to select a candidate from multiple predicates. If such category-signature based mapping is not possible, then the predicate is listed as a new relation, and the corresponding triple marked to belong to either NR-KE or NE-NE extraction class, depending on whether the target entity is already present in the KG or not.

4 Experiments

In order to evaluate effectiveness of ENTICE, we apply it to increase knowledge density for 100 randomly selected entities from each of the following five NELL categories: *Scientist, Universities, Books, Birds*, and *Cars*. For each category, a random subset of extractions in that category was evaluated using Mechanical Turk. To get a better accuracy of the evaluation, each fact was evaluated by 3 workers. Workers were made to classify each fact as correct, incorrect or can't say. Only those facts classified as correct by 2 or more evaluators were considered as correct facts.

Main Result: Experimental results comparing knowledge densities in NELL and after application of ENTICE, along with the accuracy of extractions, are presented in Table 2. From this, we observe that ENTICE is able to improve knowledge density in NELL by a factor of 7.7 while maintaining 75.5% accuracy. Sample extraction examples and accuracy per-extraction class are presented in Table 3 and Table 4, respectively.

Noun and Relation Phrase Normalization: We didn't perform any intrinsic evaluation of the entity and relation normalization step. However, in this section, we provide a few anecdotal examples to give a sense of the output quality from this step. We observe that the canopy clustering algorithm for entity and normalization is able to cluster together facts with somewhat different surface representations. For example, the algorithm came up with the following cluster with two facts: {*(J. Willard Milnor, was awarded, 2011 Abel Prize); (John Milnor, received, Abel Prize)*}. It is encouraging to see that the system is able to put *J. Willard Milnor* and *John Milnor* together, even though they have somewhat different surface forms (only one word overlap). Similarly, the relation phrases *was awarded* and *received* are also considered to be equivalent in the context of these beliefs.

Integrating with Knowledge Graph: Based on evaluation over a random-sampling, we find that entity linking in ENTICE is 92% accurate, while relation linking is about 70% accurate.

In the entity linking stage, adjectives present in a noun phrase (NP) were ignored while matching the noun phrase to entities in the knowledge graph (NELL KB in this case). In case the whole NP didn't find any match, part of the NP was used to retrieve its category, if any. For example, in *(Georg Waldemar Cantor, was born in, 1854)*, the NP *Georg Waldemar Cantor* was mapped to category *person* using his last name and *1854* to category *date*. The relation phrase *"was born in"* maps to many predicates in NELL relational metadata. NELL predicate *AtDate* was selected based on the rule that category signature of the predicate matches the category of the noun phrases present in the triple. It also has the highest frequency count for the relational phrase in the metadata.

We observed that relation mapping has lesser accuracy due to two reasons. Firstly, error in determining right categories of NPs present in a triple; and secondly, due to higher ambiguity involving relation phrases in general, i.e., a single relation phrase usually matches many relation predicates in the ontology.

5 Conclusion

This paper presents ENTICE, a simple but effective entity-centric framework for increasing knowledge densities in automatically constructed knowledge graphs. We find that ENTICE is able to significantly increase NELL's knowledge density by a factor of 7.7 at 75.5% accuracy. In addition to extracting new facts, ENTICE is also able to extend the ontology. Our goal in this paper is twofold: (1) to draw attention to the effectiveness of entity-centric approaches with bigger scope (i.e., covering all four extraction classes in Table 1) towards improving knowledge density; and (2) to demonstrate that even relatively straightforward techniques can go a long way in alleviating low knowledge density in existing state-of- the-art KGs. While these initial results are encouraging, we hope to apply ENTICE on other knowledge graphs, and also experiment with other normalization and entity linking algorithms as part of future work.

Acknowledgment

This work is supported in part by a gift from Google. Thanks to Uday Saini for carefully reading a draft of the paper.

References

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1679–1688. ACM.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues.

Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, et al. 2015. Never-ending learning. In *Proceedings of AAAI*.

Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.