An Entity-centric Approach for Overcoming Knowledge Graph Sparsity

Manjunath Hegde, Partha Talukdar
Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India
manjunath@ssl.serc.iisc.in, ppt@cds.iisc.ac.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 project 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.**

I. 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 [1], Google's Knowledge Vault [2]. 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.

Low knowledge density (or high sparsity) in automatically constructed KGs has been recognized in recent research [3]. 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 been proposed to increase knowledge density in KGs [4], [5]. 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 I. 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 I.

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

TABLE I
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.

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 I. 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 I) 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`

II. Related Work

Open Information Extraction (OIE) systems [6]–[8] 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

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

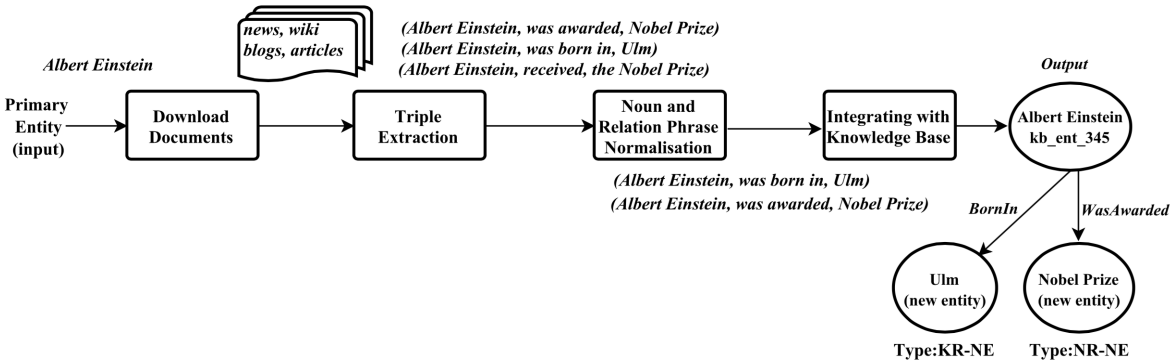

Fig. 1. Dataflow and architecture and of ENTICE. See Section III for details.

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. [9] addresses the problem of normalizing (or canonicalizing) OIE extractions which can be considered as one of the components of ENTICE (see Section III-C).
As a final step of ENTICE we map the relation to a nell-relation. There has been many efforts in all three learning paradigms (supervised, semi-supervised and unsupervised) for Relation Extraction prior to Distant Supervision. In supervised setting, sentences with labelled entities and relations are used to learn relation extractors in supervised setting. Work by [10] use supervised datasets to learn classifiers to label the relations mention between a given pair of entities in a test sentence, optionally combining relation mentions. Supervised learner suffer from problems like expensive labelled data, corpus based relation labels and classifiers tuned for given text corpus.

Other extreme for Relation Extraction is to use completely un-supervised techniques. In this approach, string of words between two entities taken from a large number of unlabelled sentences is clustered. Each cluster then signifies a relation-strings. Mapping of relation clusters to knowledge base specific relation is an added overhead in un-supervised approach. In semi-supervised learning, a very small number of seed instances or patterns are used to do bootstrap learning. [11] Original seed set is used to extract relations patterns, which further generate more instances. New instances are again used to generate relation patterns, this continues in iterative fashion. This approach suffers from low precision and semantic drift.

As previously mentioned, recent proposals for improving density of KGs such as those reported in [4], [5] focus on extracting facts of one of the four extraction classes mentioned in Table I, viz., KR-KE. The KBP challenge [12] 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 [3]. 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 the first system to the best of our knowledge that addresses the complete problem as a single framework.

III. 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.

A. Data Preprocessing

With Google API, we use RESTful requests to get web search links. Given the primary entity (PE), documents relevant to it are downloaded by issuing 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"*. Whenever available, any attribute(s) of the entity is also included as keyword. Since our method is an unsupervised way of expanding the KB, we keep the input from the user to a very minimal extent. Entity linking service is used to find the wikipedia page of the primary Entity. This helps us in disambiguating the similar entities and rejecting the wikipedia links of disambiguous entities. On an avg.top 20 documents returned by the search engine are downloaded and processed further. Number of documents are restricted by the Google API query limits. Text is extracted from the raw downloaded html documents using regex patters, HTML tag matching, and by using the Boilerpipe tool[2].

[2]Boilerpipe: http://code.google.com/p/boilerpipe

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 II
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 III
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 IV
ACCURACY BREAKDOWN OVER ENTICE EXTRACTIONS FOR EACH OF THE FOUR EXTRACTION CLASSES IN TABLE I. FOR EACH CATEGORY, APPROXIMATELY 200 EXTRACTIONS WERE EVALUATED USING MECHANICAL TURK.

B. Triple Extraction and Processing

Text of each document obtained in the previous step is processed through the Stanford CoreNLP toolkit [13]. Stanford CoreNLP provides a set of natural language analysis tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, and mark up the structure of sentences in terms of phrases and word dependencies, indicate which noun phrases refer to the same entities. We use the sentence tokenization, coreference resolution, NER and POS tags and dependency parsing features of the CoreNLP toolkit. Sentences without resolving coreference are then passed through OpenIEv4 [3] to extract *(noun phrase, predicate,noun phrase)* triples. Coref resolution is performed on these triples using the output of CoreNLP toolkit. 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 8 tokens.

The above process of representing sentence in a triple form is quite noisy. We loose lot of information when a big sentence is break down to a triple. For e.g. A sentence explaining bombing incident may contain place, time, victim, culprit, but OpenIE breaks it into many triples, each of which on their own does not describe the event completely. Example given below explains the few possibilities of noise induction.

Sentence:*In 1940 Albert Einstein renounced his German nationality law for a second time and became a U.S. Citizenship.*

OpenIE Extractions:

- 0.91747 :(Albert Einstein;renounced;his German nationality law)
- 0.91747 :(Albert Einstein;renounced;for a second time)
- 0.91747 :(Albert Einstein;renounced;In 1940)

[3]OpenIEv4: http://knowitall.github.io/openie/

- 0.95421 :(Albert Einstein;became;a U.S. Citizenship)
- 0.95421 :(Albert Einstein;became;In 1940)

Third and the last triple are capturing only time and hence miss the context. Second triple is a not very informative.

Entity Disambiguation and Linking: It is the task of determining the identity of entities mentioned in text. Named entity mentions can be highly ambiguous, any entity linking method must address this inherent ambiguity, We use dexter entity linking system which finds and mapps the entity mentions in the text to wikipedia page titles. This phase helps in clustering the different mentions of the same entity.

Extraction of Entity from Noun Phrase: Even after applying length filter, long noun phrases may not represent a real world entity. Consider the example given below,
Sentence:*Soon after their divorce, Albert Einstein married his cousin Elsa.*

- 0.91746 :(Albert Einstein;married;his cousin Elsa)
- 0.91746 :(Albert Einstein;married;Soon after their divorce)

OPenIE extraction has marked *his cousin Elsa* as the Entity-2, but we would like to have *haswife(Albert Einstein,Elsa)* as the instance of the KB. Now the challenge is how to select only "Elsa" from "his cousin Elsa". We term this as the task of obtaining entity from noun phrases. One more challenge here is to filter out the second triple because *haswife(Albert Einstein, divorce)* is an erroneous tuple.

To address both the task of obtaining entity from noun phrase and rejecting the wrong triples, we use a scoring mechanism. Scoring mechanism is based on the POS tag patterns of the entities present in wiki dump[ref??]

POS tag patterns of entities from wikipedia dump are mined and stored along with their counts. POS tag of any given noun phrase is matched with the existing POS tag patterns and the best match from phrase is selected as Entity-2. Frequency of the matched POS tag pattern, ratio of the length of Entity-2 to length of noun phrase and openIE score is used to calculate triple score.

Triple Score = Frequency of POS tag * (number of words in Entity-2/Number od words in noun phrase) * OpenIE score

C. 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 [9] 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 [14]. 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.

D. Integrating with Knowledge Graph OR Relation Mapping

1) Distance supervision: [15] proposed distant supervision for the first time in 2009. Their approach uses Freebase, a large semantic database to provide distant supervision for relation extraction. The algorithm assumes that any sentence containing two entities expresses relation between those two entities as given in freebase. Unlike supervised setting, this approach is noisy, but can learn well due to large number of examples. The algorithm depends on a database for its relation instances, and therefore does not suffer from domain-dependence or overfitting.
We use distance supervision to map the relation in a triple to a nell-relation. The distant supervision assumption is that if two entities participate in a relation, any sentence that contains those two entities might express that relation. With the above assumption, we generate training data for relation mapping using the instances from NELL. Detailed procedure for generating positive and negative training is described below. Web data is the source of distance supervision for both negative and positive training samples. For the task of relation mapping, we have hand picked 110 nell-relations. We are using the high confident nell instances and consider them as the true lables for all of the selected nell-relations. For each nell-relation we get 100 to 500 instances of the form {Entity-1:Entity-1-type, Entity-2:Entity-2-type}

[4]https://github.com/piskvorky/gensim/

For positive training samples, we take instance pairs (Entity-1,Entity-2) and using the initial part of the pipeline[??], we collect sentences from web pages. We use wildcard character '*' between entities ("Entity-1 * Entity-2") in the Google search to get more accurate sentences. These sentences contains both entities and hence represent the relation.

Even for negative training samples, we consider same set of nell relations. The generation of instances differs from positive data collection. We take Entity-1 from the instance pair and Entity-2-Neg is collected as follows, select an entity which has type same as Entity-2 and "Entity-1 nell-relation Entity-2-neg" is not present in NELL.

Example, for nell relation 'personborninlocation', NELL instance pair (Obama:person, Honolulu:location)
Sentences selected as positive training example are

- Barack Hussein Obama, born in Honolulu on August 4, 1961.
- However, a view of the various homes where Obama dwelled in Honolulu reveals a broader, diversified beginning.

Examples for negative training.

- Obama Pokes Fun at Friends and Enemies at dinner held in Washington.
- Obama walks to stage at rally in Washington.

There is always a noise factor associated with distance supervision and it affects the accuracy of the system.

2) CNN for Relation Mapping:: High level architecture of the CNN is shown in Figure 2 For relation linking, the distantly supervised training data is used to train a classification model. Given an instance, which consists of a plain text sentence, an entity pair, and the NELL type of the entities, the model generates a probability distribution over the candidate relations.

The classification model used is a Convolutional Neural Network (CNN) following [16]. In this model, the input instance is represented by an $n \times d$ *image*, which is then fed through a CNN. Two filters (of height 1 and 2) are used in a single convolution layer, followed by a max-pooling layer. These are 1D (temporal) convolutions, and therefore the width of the filter is equal to the width of the image. The height, therefore, is analogous to the size of a context window around tokens in a sentence.

To convert the raw text into an input image, the following features are used. Each feature is embedded using a matrix of parameters that is learned as part of the training process.

- Word features: word ids are embedded using an embedding matrix that is initialized using Glove [17] vectors.
- POS tags: Generated using CoreNLP
- NELL type information: Generated using ???
- Position features: From [**?**], these features are integers that represent the relative distance of each token from an anchor entity (see Fig.). Two such features are generated, one for each entity in the entity pair. These features are useful in incorporating temporal information about word-ordering into the input, which otherwise a CNN is agnostic to.

d is therefore $d_{word} + d_{pos-tag} + d_{NELL-type} + 2 \times d_{position}$). d_{word} is set to 300, while all the other embedding matrices are chosen do be 50 dimensional. The number of filters used is 256.

3) Relatiom Mapping Using Extraction Patterns: The set of normalized triples from the previous step are linked with the Knowledge Graph, whenever possible, in this step. Metadata for each relation in NELL has a set of verb phrases, which capture the variation of the nell-relation in free text. The similarity of words in triple to avg similarity of words in the extraction pattern is used for relation mapping. GloVe vector representation of words is the key behind calculating the similarity. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus. Each word is represented as a vector of dimension 300. The algorithm 1 describes all the steps of relation mapping. Whenever this extraaction pattern 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 NR-NE extraction class, depending on whether the target entity is already present in the KG or not.

IV. 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 II. 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 III and Table IV, 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

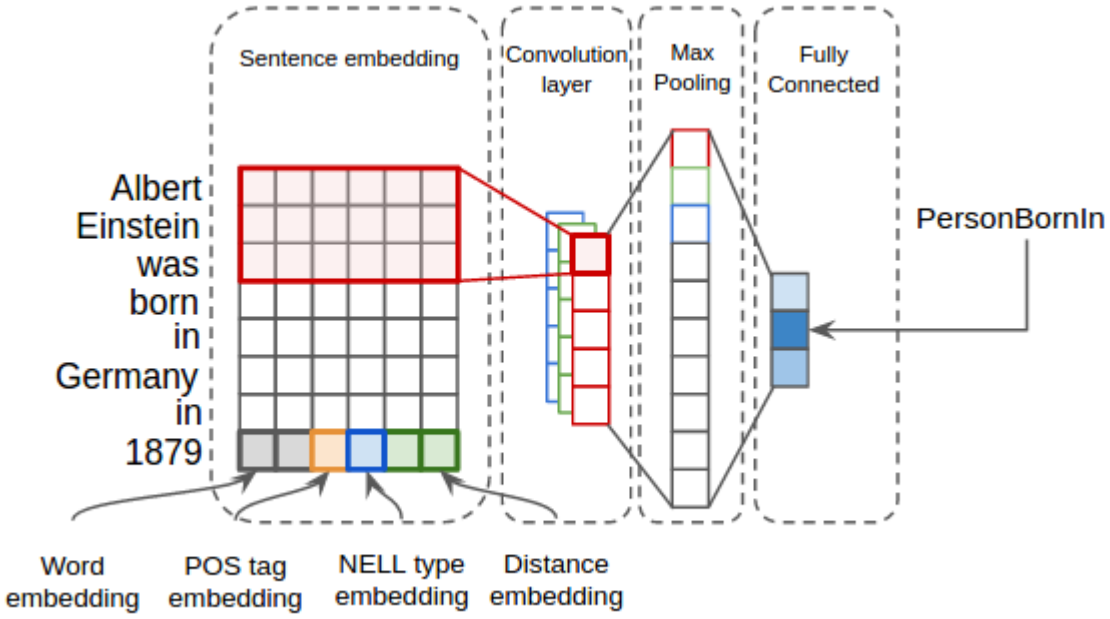

Fig. 2. CNN for relation mapping. See Section III-D for details.

Algorithm 1 Relation Mapping with Extraction Patterns and Glove Vectors

procedure RELATION MAPPING
  Triple: *Entity-1,relation,Entity-2*
  get entity type (*Entity-1-Type,Entity-2-Type*) using NELL Json API
  relation vector (RV) = initialise 0-vector of length 300
  for each word in relation **do**
    RV = RV + GloVe(word)
  end for
  RV = RV/number of words in relation
  S = Set(nell-relations which satisy type constraints)
  for each nell-relation in S **do**
    P=Extraction patterns for nell-relation
    for each extraction-pattern in P **do**
      EPV = initialise zero vector
      for each word in extraction pattern **do**
        EPV = EPV + GloVe(word)
      end for
      EPV = EPV/number of words in extraction pattern
    end for
    similarity = cosineSimilarity(RV,EPV)
  end for
  Mapped Relation = relation with maximum similarity score
end procedure

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.

V. 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. Future work will include noise reduction in the pipeline. Close examining of sentences while constructing triples to make sure that noisy triples are not added to the system.

ENTICE can be applied to other knowledge graphs with appropriate changes. Experiment with other normalization and relation mapping algorithms are the part of future work.

Acknowledgment

I would like to thank Dr. Partha Talukdar for his valuable guidance and support throught the project. Special thanks to Tushar Nagarajan for building the CNN model and helping me throught the relation mapping process. All the other members of the Machine and Language Learning (MALL) lab friends for their suggestions and timely help. I thank MALL lab and Computational and Data Science Department for providing Computational facilities.

References

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

[2] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "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*, 2014.

[3] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge base completion via search-based question answering," in *Proceedings of the 23rd international conference on World wide web*, 2014.

[4] M. Gardner, P. P. Talukdar, B. Kisiel, and T. Mitchell, "Improving learning and inference in a large knowledge-base using latent syntactic cues," 2013.

[5] M. Gardner, P. P. Talukdar, J. Krishnamurthy, and T. Mitchell, "Incorporating vector space similarity in random walk inference over knowledge bases," 2014.

[6] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "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*. Association for Computational Linguistics, 2007, pp. 25–26.

[7] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[8] M. Schmitz, R. Bart, S. Soderland, O. Etzioni *et al.*, "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*. Association for Computational Linguistics, 2012, pp. 523–534.

[9] L. Galárraga, G. Heitz, K. Murphy, and F. M. Suchanek, "Canonicalizing open knowledge bases," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 1679–1688.

[10] G. ZHOU12, M. Zhang, D. H. Ji, and Q. Zhu, "Tree kernel-based relation extraction with context-sensitive structured parse tree information," *EMNLP-CoNLL 2007*, p. 728, 2007.

[11] B. Rozenfeld and R. Feldman, "Self-supervised relation extraction from the web," *Knowledge and Information Systems*, vol. 17, no. 1, pp. 17–33, 2008.

[12] M. Surdeanu, "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)*, 2013.

[13] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[15] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[16] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162