DeepLENS: Deep Learning for Entity Summarization⋆

Qingxia Liu, Gong Cheng, and Yuzhong Qu

National Key Laboratory for Novel Software Technology, Nanjing University, China
qxliu2013@smail.nju.edu.cn, {gcheng,yzqu}@nju.edu.cn

Abstract. Entity summarization has been a prominent task over knowledge graphs. While existing methods are mainly unsupervised, we present DeepLENS, a simple yet effective deep learning model where we exploit textual semantics for encoding triples and we score each candidate triple based on its interdependence on other triples. DeepLENS significantly outperformed existing methods on a public benchmark.

1 Introduction

Entity summarization is the task of computing a compact summary for an entity by selecting an optimal size-constrained subset of entity-property-value triples from a knowledge graph such as an RDF graph [7]. It has found a wide variety of applications, for example, to generate a compact entity card from Google's Knowledge Graph where an entity may be described in dozens or hundreds of triples. Generating entity summaries for general purposes has attracted much research attention, but existing methods are mainly unsupervised [2,9,3,4,13,10,6,5,11]. One research question that naturally arises is *whether deep learning can much better solve this task.*

To the best of our knowledge, ESA [12] is the only supervised method in the literature for this task. ESA encodes triples using graph embedding (TransE), and employs BiLSTM with supervised attention mechanism. Although it outperformed unsupervised methods, the improvement reported in [12] was rather marginal, around $+7\%$ compared with unsupervised FACES-E [4] on the ESBM benchmark [8]. It inspired us to explore more effective deep learning models for the task of general-purpose entity summarization.

In this short paper, we present DeepLENS,[1] a novel **Deep L**earning based approach to **EN**tity **S**ummarization. DeepLENS uses a simple yet effective model which addresses the following two limitations of ESA, and thus achieved significantly better results in the experiments.

1. Different from ESA which encodes a triple using graph embedding, we use word embedding because we consider textual semantics more useful than graph structure for the entity summarization task.

---

[1] https://github.com/nju-websoft/DeepLENS

2. Whereas ESA encodes a set of triples as a sequence and its performance is sensitive to the chosen order, our aggregation-based representation satisfies permutation invariance and hence more suitable for entity summarization.

In the remainder of the paper, Section 2 details DeepLENS, Section 3 presents experiment results, and Section 4 concludes the paper.

2 Approach

Problem Statement An RDF graph T is a set of triples. The *description* of entity e in T, denoted by $\mathtt{Desc}(e) \subseteq T$, comprises triples where e is the subject or object. Each triple $t \in \mathtt{Desc}(e)$ describes a property $\mathtt{prop}(t)$ which is the predicate of t, and gives a value $\mathtt{val}(t)$ which is the object or subject of t other than e. For a size constraint k, a *summary* of e is a subset of triples $S \subseteq \mathtt{Desc}(e)$ with $|S| \leq k$. We aim to generate an optimal summary for general purposes.

Overview of DeepLENS Our approach DeepLENS generates an optimal summary by selecting k most salient triples. As a supervised approach, it learns salience from labeled entity summaries. However, two issues remain unsolved. First, knowledge graph such as RDF graph is a mixture of graph structure and textual content. The effectiveness of a learning-based approach to entity summarization relies on a *proper representation of entity descriptions of such mixed nature*. Second, the salience of a triple is not absolute but dependent on the context, i.e., the set of other triples in the entity description. It is essential to *represent their independence*. DeepLENS addresses these issues with the scoring model presented in Fig. 1. It has three modules which we will detail below: triple encoding, entity description encoding, and triple scoring. Finally, the model scores each candidate triple $t \in \mathtt{Desc}(e)$ in the context of $\mathtt{Desc}(e)$.

Triple Encoding For entity e, a triple $t \in \mathtt{Desc}(e)$ provides a property-value pair $\langle \mathtt{prop}(t), \mathtt{val}(t) \rangle$ of e. Previous research [12] leverages graph embedding to encode the structural features of $\mathtt{prop}(t)$ and $\mathtt{val}(t)$. By contrast, for the task of entity summarization we consider textual semantics more important than graph structure, and we *solely exploit textual semantics* for encoding t.

Specifically, for RDF resource r, we obtain its *textual form* as follows. For an IRI or a blank node, we retrieve its `rdfs:label` if it is available, otherwise we have to use its local name; for a literal, we take its lexical form. We represent each word in the textual form by a pre-trained word embedding vector, and we average these vectors over all the words to represent r, denoted by $\mathrm{Embedding}(r)$. For triple $t \in \mathtt{Desc}(e)$, we generate and concatenate such vector representations for $\mathtt{prop}(t)$ and $\mathtt{val}(t)$ to form $\boldsymbol{t}$, the *initial representation* of t. Then $\boldsymbol{t}$ is fed into a multi-layer perceptron (MLP) to generate $\boldsymbol{h}$, the *final representation* of t:

$$\boldsymbol{t} = [\mathrm{Embedding}(\mathtt{prop}(t)); \ \mathrm{Embedding}(\mathtt{val}(t))] \,, \quad \boldsymbol{h} = \mathrm{MLP_C}(\boldsymbol{t}) \,. \quad (1)$$
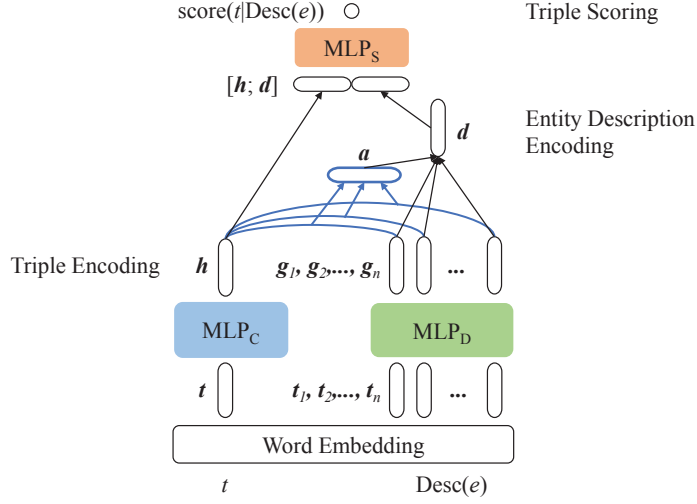
Fig. 1. Model of DeepLENS.

Entity Description Encoding To score a candidate triple in the context of other triples in the entity description, previous research [12] captures the independence between triples in $\texttt{Desc}(e)$ using BiLSTM to pass information. Triples are fed into BiLSTM as a sequence. However, $\texttt{Desc}(e)$ is a set and the triples lack a natural order. The performance of this model is unfavourably sensitive to the order of input triples. Indeed, as we will show in the experiments, different orders could lead to considerably different performance.

To generate a representation for $\texttt{Desc}(e)$ that is *permutation invariant*, we perform aggregation. Specifically, let $t_1, \ldots, t_n$ be the initial representations of all the n triples in $\texttt{Desc}(e)$ computed by Eq. (1). We feed a MLP with each t_i for $1 \leq i \leq n$ and generate their final representations $g_1, \ldots, g_n$, which in turn are weighted using attention mechanism from h computed by Eq. (1), the final representation of the candidate triple t to be scored. We calculate the sum of these weighted representations of triples to represent $\texttt{Desc}(e)$, denoted by d:

$$g_i = \text{MLP}_{\text{D}}(t_i), \quad a_i = \frac{\exp(\cos(h, g_i))}{\sum_j \exp(\cos(h, g_j))}, \quad d = \sum_{i=1}^{n} a_i g_i. \tag{2}$$

where the cos function computes the cosine similarity between two vectors, and a_i is the i-th component of the attention vector a. The result of summation is not sensitive to the order of triples in $\texttt{Desc}(e)$.

Triple Scoring For each candidate triple $t \in \texttt{Desc}(e)$ to be scored, we concatenate its final representation h and the representation d for $\texttt{Desc}(e)$. We feed the result into a MLP to compute the context-based salience score of t:

$$\texttt{score}(t|\texttt{Desc}(e)) = \text{MLP}_{\text{S}}([h; \; d]). \tag{3}$$

Parameters of the entire model are jointly trained based on the mean squared error loss, supervised by labeled entity summaries.

3   Experiments

3.1   Datasets

We used ESBM v1.2, the largest available benchmark for evaluating general-purpose entity summarization [8].[2] For each of 125 entities in DBpedia and 50 entities in LinkedMDB, this benchmark provided 6 ground-truth summaries created by different human experts under $k = 5$, and another 6 ground-truth summaries under $k = 10$. We used the train-valid-test split specified in the benchmark to perform five-fold cross-validation.

3.2   Participating Methods

We compared DeepLENS with 10 baseline methods.

Unsupervised Methods. We compared with 9 unsupervised methods that had been tested on ESBM: RELIN [2], DIVERSUM [9], FACES [3], FACES-E [4], CD [13], LinkSUM [10], BAFREC [6], KAFCA [5], and MPSUM [11]. We directly presented their results reported on the ESBM website.

Supervised Methods. We compared with ESA [12], the only supervised method in the literature to our knowledge. We reused its open-source implementation and configuration.[3] We fed it with triples sorted in alphabetical order.

For our approach DeepLENS, we used 300-dimensional fastText [1] word embedding vectors trained on Wikipedia to generate initial representations of triples. The sizes of fully connected layers in MLP_C, MLP_D, and MLP_S were {64, 64}, {64, 64}, and {64, 64, 64, 1}, respectively. All hidden layers used ReLU as activation function. In particular, the output layer of the entire model, i.e., the last layer of MLP_S, consisted of one linear unit. We trained the model using Adam optimizer with learning rate 0.01.

For both ESA and DeepLENS, we performed early stopping on the validation set to choose the number of training epochs from 1–50.

Oracle Method. ORACLE approximated the best possible performance on ESBM and formed a reference point used for comparisons [8]. It outputted k triples that most frequently appeared in ground-truth summaries.

3.3   Results

Following ESBM, we compared machine-generated summaries with ground-truth summaries by calculating F1 score, and reported the mean F1 achieved by each method over all the test entities in a dataset.

Table 1. Average F1 over all the test entities. Significant and insignificant differences ($p < 0.01$) between DeepLENS and each baseline are indicated by ▲ and ○, respectively.

	DBpedia		LinkedMDB	
	$k = 5$	$k = 10$	$k = 5$	$k = 10$
RELIN [2]	0.242	0.455	0.203	0.258
DIVERSUM [9]	0.249	0.507	0.207	0.358
FACES [3]	0.270	0.428	0.169	0.263
FACES-E [4]	0.280	0.488	0.313	0.393
CD [13]	0.283	0.513	0.217	0.331
LinkSUM [10]	0.287	0.486	0.140	0.279
BAFREC [6]	0.335	0.503	0.360	0.402
KAFCA [5]	0.314	0.509	0.244	0.397
MPSUM [11]	0.314	0.512	0.272	0.423
ESA [12]	0.331	0.532	0.350	0.416
DeepLENS	0.404 ▲▲▲▲▲▲▲▲▲	0.575 ▲▲▲▲▲▲▲▲▲	0.469 ▲▲▲▲▲▲▲▲▲	0.489 ▲▲▲▲▲▲▲▲▲
ORACLE	0.595	0.713	0.619	0.678

Table 2. Average F1 over all the test entities achieved by different variants of ESA.

	DBpedia		LinkedMDB	
	$k = 5$	$k = 10$	$k = 5$	$k = 10$
ESA	0.331	0.532	0.350	0.416
ESA-text	0.379	0.558	0.390	0.418
ESA-rnd	$0.116_{\pm 0.008}$	$0.222_{\pm 0.007}$	$0.113_{\pm 0.015}$	$0.219_{\pm 0.011}$

Comparison with Baselines. As shown in Table 1, supervised methods were generally better than unsupervised methods. Our DeepLENS outperformed all the baselines including ESA. Moreover, two-tailed t-test showed that all the differences were statistically significant ($p < 0.01$) in all the settings. DeepLENS achieved new state-of-the-art results on the ESBM benchmark. However, the notable gaps between DeepLENS and ORACLE suggested room for improvement and were to be closed by future research.

Ablation Study. Compared with ESA, we attributed the better performance of DeepLENS to two improvements in our implementation: the exploitation of textual semantics, and the permutation invariant representation of triple set. They were demonstrated by the following ablation study of ESA.

First, we compared two variants of ESA by encoding triples in different ways. For triple t, the original version of ESA encoded the structural features of $\text{prop}(t)$ and $\text{val}(t)$ using TransE. We implemented ESA-text, a variant that encoded both $\text{prop}(t)$ and $\text{val}(t)$ using fastText as in our approach. As shown in Table 2, ESA-text slightly outperformed ESA, showing the usefulness of textual semantics compared with graph structure used by ESA.

Second, we compared two variants of ESA by feeding with triples in different orders. The default version of ESA was fed with triples sorted in alphabetical order for both training and testing. We implemented ESA-rnd, a variant that was fed with triples in alphabetical order for training but in random order for testing. We tested ESA-rnd 20 times and reported its mean F1 with standard deviation.

In Table 2, the notable falls from ESA to ESA-rnd showed the unfavourable sensitivity of BiLSTM used by ESA to the order of input triples.

4   Conclusion

We presented DeepLENS, a simple yet effective deep learning model for general-purpose entity summarization. It has achieved new state-of-the-art results on the ESBM benchmark, significantly outperforming existing methods. Thus, entity summarization becomes another research field where a combination of deep learning and knowledge graph is likely to shine. However, in DeepLENS we only exploit textual semantics. In future work, we will incorporate ontological semantics into our model. We will also revisit the usefulness of structural semantics.

Acknowledgments

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. TACL **5**, 135–146 (2017)
2. Cheng, G., Tran, T., Qu, Y.: RELIN: relatedness and informativeness-based centrality for entity summarization. In: ISWC 2011, Part I. pp. 114–129 (2011)
3. Gunaratna, K., Thirunarayan, K., Sheth, A.P.: FACES: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In: AAAI 2015. pp. 116–122 (2015)
4. Gunaratna, K., Thirunarayan, K., Sheth, A.P., Cheng, G.: Gleaning types for literals in RDF triples with application to entity summarization. In: ESWC 2016. pp. 85–100 (2016)
5. Kim, E.K., Choi, K.S.: Entity summarization based on formal concept analysis. In: EYRE 2018 (2018)
6. Kroll, H., Nagel, D., Balke, W.T.: BAFREC: Balancing frequency and rarity for entity characterization in linked open data. In: EYRE 2018 (2018)
7. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: Entity summarization: State of the art and future challenges. CoRR **abs/1910.08252** (2019)
8. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: ESBM: An entity summarization benchmark. In: ESWC 2020 (2020)
9. Sydow, M., Pikula, M., Schenkel, R.: The notion of diversity in graphical entity summarisation on semantic knowledge graphs. J. Intell. Inf. Syst. **41**(2), 109–149 (2013)
10. Thalhammer, A., Lasierra, N., Rettinger, A.: LinkSUM: Using link analysis to summarize entity data. In: ICWE 2016. pp. 244–261 (2016)
11. Wei, D., Gao, S., Liu, Y., Liu, Z., Huang, L.: MPSUM: Entity summarization with predicate-based matching. In: EYRE 2018 (2018)
12. Wei, D., Liu, Y., Zhu, F., Zang, L., Zhou, W., Han, J., Hu, S.: ESA: Entity summarization with attention. In: EYRE 2019. pp. 40–44 (2019)
13. Xu, D., Zheng, L., Qu, Y.: CD at ENSEC 2016: Generating characteristic and diverse entity summaries. In: SumPre 2016 (2016)