ESLM: Improving Entity Summarization by Leveraging Language Models

Asep Fajar Firmansyah[1,2(✉)], Diego Moussallem[1,3], and Axel-Cyrille Ngonga Ngomo[1]

[1] Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany
{diego.moussallem,axel.ngonga}@upb.de
[2] The State Islamic University Syarif Hidayatullah Jakarta, Jakarta, Indonesia
asep.fajar.firmansyah@upb.de, asep.airlangga@uinjkt.ac.id
[3] Jusbrasil, Salvador, Brazil

Abstract. Entity summarizers for knowledge graphs are crucial in various applications. Achieving high performance on the task of entity summarization is hence critical for many applications based on knowledge graphs. The currently best performing approaches integrate knowledge graphs with text embeddings to encode entity-related triples. However, these approaches still rely on static word embeddings that cannot cover multiple contexts. We hypothesize that incorporating contextual language models into entity summarizers can further improve their performance. We hence propose ESLM (Entity Summarization using Language Models), an approach for enhancing the performance of entity summarization that integrates contextual language models along with knowledge graph embeddings. We evaluate our models on the datasets DBpedia and LinkedMDB from ESBM version 1.2, and on the FACES dataset. In our experiments, ESLM achieves an F-measure of up to 0.591 and outperforms state-of-the-art approaches in four out of six experimental settings with respect to the F-measure. In addition, ESLM outperforms state-of-the-art models in all experimental settings when evaluated using the NDCG metric. Moreover, contextual language models notably enhance the performance of our entity summarization model, especially when combined with knowledge graph embeddings. We observed a notable boost in our model's efficiency on DBpedia and FACES. Our approach and the code to rerun our experiments are available at https://github.com/dice-group/ESLM.

Keywords: Entity Summarization · Language Models · Knowledge Graph Embeddings

1 Introduction

Entity summarizers are extensively utilized across user-facing applications driven by knowledge graphs (e.g., Web search [10], RDF browsers [7], and recommender systems [22]) to provide succinct summaries of entities, and hence facilitate user

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Meroño Peñuela et al. (Eds.): ESWC 2024, LNCS 14664, pp. 162–179, 2024.
https://doi.org/10.1007/978-3-031-60626-7_9

comprehension. Recent methods (e.g., ESA [28], DeepLENS [15], GATES [8], and ESCS [4]) have achieved improved effectiveness by employing deep learning algorithms to encode triples containing entity descriptions and generate accurate summaries. For instance, ESA combines word embeddings [2] with Knowledge Graph Embeddings (KGEs) (e.g., computed using TransE [3]) to transform the predicate and object of each triple into vectors. By combining word embeddings and KGEs, ESA significantly outperforms previous methods based on unsupervised learning [5,9,25,27].

While approaches based on embeddings are effective, they exhibit an important limitation: They all rely on static word embeddings such as Word2Vec [2] and GloVe [20]. These static models are unable to account for the various contexts of a word, particularly in cases where homonyms appear in different entity descriptions. The motivation behind our work was hence to validate the following hypothesis: *Integrating contextual language models (LMs) into entity summarization methods can enhance their performance.* We hence present ESLM (Entity Summarization using Language Models), an entity summarization that leverages contextual LMs alongside the topological information of KGEs. In our implementation of ESLM, we use the contextual LMs BERT [6] and ERNIE [23] because of the differing representations they compute: BERT acquires dynamic representations based on the transformer architecture using purely textual data, while ERNIE enhances these representations with external knowledge from KGs [32]. Additionally, ESLM incorporates a transformed-based large LM based on the T5 model [21]. We conduct a comprehensive evaluation of ESLM using the ESBM (Entity Summarization BenchMark, version 1.2) dataset [13], which comprises datasets based on DBpedia and LinkedMDB. We also evaluate our approach on the FACES [9] dataset. ESLM consistently outperforms the current state-of-the-art (SOTA) methods on these datasets w.r.t. the normalized discounted cumulative gain (NDCG) measure. Additionally, ESLM achieves an F-measure of up to 0.591 on the DBpedia dataset.

Our contributions to entity summarization are as follows:

- We introduce a new approach for entity summarization dubbed ESLM, which leverages contextual LMs combined with KGE to enhance the performance of entity summarization.
- We conduct a detailed ablation study to analyze the impact of contextual LMs and their integration with KGE within our model. This study aids in understanding the contribution of key components to the overall performance of ESLM.
- We conclude from our findings that the utilization of contextual LMs and their integration with KGE significantly outperforms SOTA, particularly on the DBpedia and FACES datasets.

The rest of this paper is organized as follows: Sect. 2 provides a summary of related works. Section 3 defines the entity summarization problem formally and introduces our ESLM model. Section 4 details the implementation of our approach. Our evaluation is described in Sect. 5. Finally, Sect. 6 concludes the paper and suggests directions for future work.

2 Related Work

2.1 Entity Summarization

Unsupervised learning has previously been used in entity summarization tasks on single and combined features [14] such as frequency, centrality, informativeness, and similarity. For example, RELIN [5] computes the relatedness between RDF triples and uses the informativeness measurement of each triple in a random surfer model. DIVERSUM [24] employs the concept of diversification to address the entity summarization problem, incorporating it into its summarizing algorithm. FACES [9] utilizes all the aforementioned dimensional features to generate entity summaries.

Recently, deep learning techniques are being used for entity summarization. ESA [28], the first model to use deep learning for this purpose, employs bidirectional long short-term memory (BiLSTM) networks. It combines a word embedding technique [2] with TransE [3] to encode the predicate and object of a triple, identifying these components as crucial for summarizing an entity's triples [27]. ESA applies BiLSTM with an attention mechanism, selecting the top-k triples for the entity summary. In contrast, DeepLENS [15] relies solely on word embeddings, specifically fastText [11] to encode triples containing text-based entity descriptions. The authors argue that word embeddings provide richer textual semantics than KGEs for this task. They proceed to show that DeepLENS outperforms ESA on benchmark datasets like ESBM (version 1.2) [13]. GATES [8] combines GloVe word embeddings [20] with KGE (such as ComplEx [26]) using graph neural networks (GNNs). This method aims to enhance the quality of entity summaries by encoding topological information through KGE. GATES outperforms both DeepLENS and ESA on the ESBM (version 1.2) and FACES datasets. Most recently, ESCS [4] was introduced. It employs an approach similar to DeepLENS by using Word2Vec [18], and introduces a novel method for the computation of triple scores and the construction of summary sets for any target entity. This method is based on the idea of salience, which is computed by evaluating the similarity between an entity's semantic embeddings and a particular property (predicate). Additionally, it computes and exploits the complementarity of predicates and aims to optimise the complementary of the relationships it returns in entity summaries.

2.2 Contextual Language Models

In recent years, contextual LMs have significantly impacted various downstream tasks, such as question answering (QA) [31], text summarization [17,19], and relation extraction [1]. These models achieve the current SOTA performance in numerous Natural Language Processing (NLP) tasks. Contextual LMs, as opposed to static word embeddings, provide each token with a representation derived from the entire input sequence. This approach allows them to capture more contextual information than static embeddings. For example, BERT [6] is a pre-trained contextual LM that is built upon a multi-layer bidirectional

transformer encoder. The model is trained on a corpus that includes the English Wikipedia and the BooksCorpus. During training, BERT employs a masked language modeling technique, wherein certain tokens within an input sequence are randomly replaced with a `[MASK]` token. BERT consequently learns to predict these masked tokens based on the context provided by the unmasked tokens in the sequence. To further enhance LMs, ERNIE [32] incorporates knowledge graphs into the computation of word embeddings. ERNIE constructs entity representations from words, encoding them using KGE models such as TransE. Similarly to BERT, it then utilizes masked LMs and next-sentence prediction for pre-training, and for the extraction of lexical and syntactic information from text tokens. Like BERT, ERNIE uses English Wikipedia for pre-training and aligns text with Wikidata through KGE.

Unlike BERT and ERNIE, T5 models every NLP problem as a text-to-text problem [21]. The model is trained using a denoising autoencoder objective, where it learns to reconstruct the original text from a corrupted version. The T5 model does not require task-specific heads. It distinguishes tasks in the input text through prefixes that guide the model during the output computation and generation. For example, the user may provide an input in the form of `classify: text` to prompt the model to output a class label for the input text.

3 Approach

3.1 Problem Statement

Entity Description. Let E be a set of entities, R be a set of relations, C be a set of classes and L denote a set of literals. A knowledge graph $T \subseteq E \times R \times (C \cup L \cup E)$ is a set of triples (s, p, o). s is called the subject, p the predicate, and o the object of the triple. We define an entity description, $Desc(e, T)$, as follows:

$$Desc(e, T) = \{(e, r, o) \in T \vee (s, r, e) \in T\} \tag{1}$$

where $s, e \in E$, $o \in (C \cup L \cup E)$, and r is a predicate. An example of such as description is provided in Fig. 1. Note that the triples (`3WAY FM`, `Type`, `Radio Station`) and (`Warrnambool`, `Broadcast Area of`, `3WAY FM`) belong to the description of `3WAY FM`, i.e., as per Eq. 1, the target entity e can function as either a subject or an object in the elements of its description $Desc(e, T)$.

Entity Summarization. Let e be an entity e, $Desc(e, T)$ be its entity description, and $k \in \mathbb{N}$ be a size constraint. We define an entity summary $ES(e)$ as a subset of $Desc(e, T)$ with $|ES(e)| \leq k$, where $k = 5$ or $k = 10$ is often used in practice. The purpose of entity summarization techniques is to compute $ES(e)$ by selecting the k best suited triples from $Desc(e, T)$.

3.2 ESLM Model

The ESLM relies on a Transformer-based language model [16], allowing for context-aware processing and prediction. The attention mechanism allows the

model to focus on the most relevant aspects of data sequences. Additionally, ESLM employs a multi-layer perceptron (MLP) for accurate triple scoring, which aids in selecting the most relevant triples for each entity. Moreover, we enrich the model with KGEs to augment the model's effectiveness, leveraging the rich semantic information from the knowledge graphs. The architecture of ESLM is detailed in Fig. 1. In the following subsections, we discuss each of our model's components.

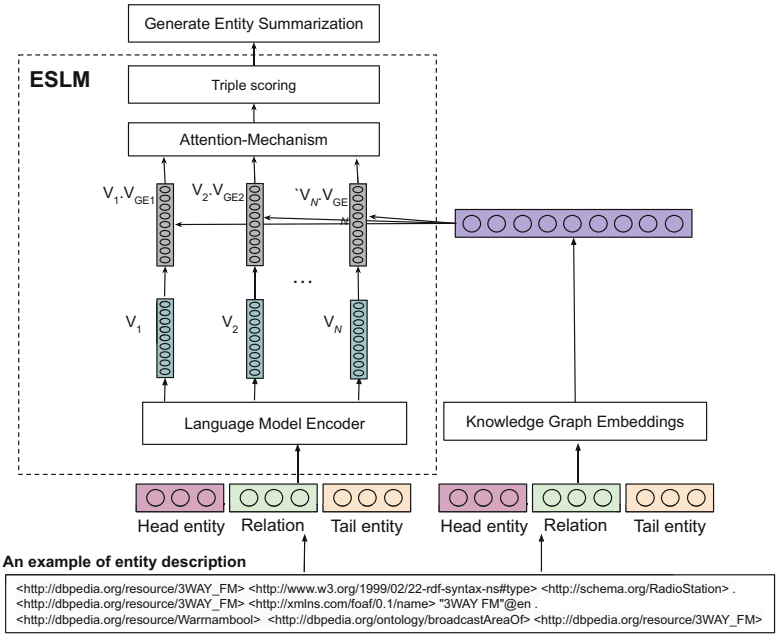

Fig. 1. The ESLM model architecture

Language Model Encoder. This encoder relies on a pre-trained model based on the Transformer architecture (e.g., BERT, ERNIE, or T5), which is further fine-tuned using labeled data from the input representations. The encoder's configuration includes the number of layers (i.e., transformer blocks) denoted by N_L set to 12, the number of hidden layers marked by H set to 768, and the number of self-attention heads denoted by AH set to 12.

Each component (subject (s), predicate (p), object (o)) from the triple t is represented in their textual form. When RDF resources are identified as IRIs (Internationalized Resource Identifiers), we utilize their `rdfs:label` for textual representation; otherwise, the local name of the IRI is used as the textual representation. The local name, extracted from the IRI segment after the last slash ('/'), acts as a unique, concise identifier within the IRI's namespace, serving as a

human-readable term when a label is absent. Using the text form allows the LM to process and vectorize IRIs effectively, ensuring that the lexical and semantic nuances of the triples are captured during encoding. As proposed in [30], each triple t within the entity description $Desc(e, T)$ is structured as a single sequence-packed sentence (s, p, o). Furthermore, all these single sequences are formatted as input representations for the ESLM model. Every sequence begins with the special classification token (`[CLS]`). Additionally, we use a special separator token (`[SEP]`) to distinctly separate each component of the triple. For example, a triple t such as (`3WAY FM`, `Type`, `Radio Station`) is transformed into textual input such as (`3WAY FM [SEP] Type [SEP] Radio Station`) and then translated into a series of tokens denoted as $\{Tok_1, Tok_2, \ldots, Tok_N\}$, where N is the count of tokens. Furthermore, the input tokens of t are converted into embeddings by implementing a transformer-based encoder. Moreover, the LM encoder generates two kinds of outputs. The first is the final hidden state of the special classification token, denoted as (`[CLS]`) and represented by $C \in \mathbb{R}^H$, which is an aggregate representation of the entire input sequence often used in classification tasks. The second output consists of the last hidden states for each input token in the sequence. These are denoted as V_i, leading to set a $V = \{V_1, V_2, \ldots, V_N\}$, where each $V_i \in \mathbb{R}^H$ corresponds to the i^{th} input token. Here, N is the sequence length, and H is the hidden state dimension.

Attention Mechanism. In the ESLM model, the attention mechanism is applied through a three-step process to assess the importance of each token in the LM encoder output. Initially, attention scores are assigned to each token using a linear transformation represented by $A_{weights} = VW_{attn} + b_{attn}$. Here, V is the LM encoder output. Meanwhile, W_{attn} and b_{attn} are the weights and bias of the linear transformation, which are hence normalized to form a valid probability distribution. The Softmax function is then applied to transform these scores into probabilities that sum up to 1, reflecting the relative significance of each token in the sequence. This mechanism enables the model to focus selectively on the most relevant parts of the input, enhancing its language understanding and generation capabilities. The complete equation is shown in Eq. 2:

$$A_{weights} = Softmax(VW_{attn} + b_{attn}). \tag{2}$$

Subsequently, these normalized scores are applied to the output V of the LM encoder using element-wise multiplication, which is shown by $\odot$, resulting in the attention-weighted output A, which can be seen in Eq. 3:

$$A = A_{weights} \odot V. \tag{3}$$

This process allows the model to focus selectively on the most relevant parts of the input sequence, enhancing its ability to interpret and generate language by considering both the context and significance of each element in the sequence.

Triple Scoring. This step relies on the MLP, which is applied to learn the output from the attention mechanism process A into a form that is more amenable for the subsequent operations—calculating the mean and applying the softmax function. The mean operation suggests an aggregation or summarization of the features learned by the MLP, and the softmax function indicates that the final goal might be to interpret these aggregated features probabilistically, possibly for a task like regression or probabilistic classification. The scoring function for a triple is denoted S_t, where $S_t \in \mathbb{R}$, is calculated as shown in Eq. 4:

$$S_t = softmax(mean(MLP(A))). \tag{4}$$

3.3 ESLM Model Enrichment

Figure 1 demonstrates that the triple-scoring computation is based on semantic information generated by a pre-trained LM (PLM) that does not consider the structural information of KGs. We now enhance our model by incorporating KGEs based on the ComplEx method to obtain information from the structure information of the KG.

KGE Using ComplEx. Let $GE : E \cup R \cup L \cup C \rightarrow \mathbb{C}^d$ be an embedding function that computes vectors for the elements of a knowledge graph. In this work, we use ComplEx with $d = 300$, i.e., every component of a triple (s, p, o) is represented by a 300-dimensional complex-valued vector. The vector for a given triple, $V_{GE_{ti}}$, is constructed by concatenating the embeddings for each component of the triple (s_{ti}, p_{ti}, o_{ti}), thus creating a single vector that holds all the structural information for that triple, as indicated in Eq. 5:

$$V_{GE_{ti}} = GE(s_{ti})||GE(p_{ti})||GE(o_{ti}). \tag{5}$$

Integration of LM with KGE. To leverage both structured and unstructured information, the model concatenates the last hidden vectors V_{ti} from the LM encoder with the KGE vectors $V_{GE_{ti}}$, resulting in a new vector V'_{ti} as shown in Eq. 6. This concatenated vector V'_{ti} holds both the contextualized semantic information from the LM and the structural relationships from the KG, yielding a more comprehensive representation for each triple:

$$V'_{ti} = V_{ti}||V_{GE_{ti}}. \tag{6}$$

By concatenating these vectors, we literally fuse textual information (V_{ti}) with knowledge graph information ($V_{GE_{ti}}$). An attention mechanism is applied to the concatenated vectors V'_{ti} to focus on the most relevant parts of the combined embedding while generating the summary. Therewith, the model learns the relation between the embedding dimensions in LM and KGE representations. As shown in our experiments, this enables our model to better discern the salience of triples for entities to summarise. The exact computation of this operation is given by the formulas outlined in Eqs. 2 and 3, and involve calculating attention

weights and context vectors. With the attention-enriched vectors V', the model computes the triple scoring as specified in Eq. 4. The scoring process evaluates the relevance of each triple for the summary using the enriched embeddings that now incorporate both semantic and structural insights.

3.4 Entity Summarization

Finally, the entity summary $ES(e)$ is generated for an entity e by selecting the top-k triples from $Desc(e, T)$, which is a set of entity descriptions for entity e in a KG. The parameter k represents a size constraint, indicating the number of triples to be included in the summary. The selection is based on Eq. 7, as defined by [8].

$$\forall t' \in ES(e) \backslash Desc(e, T) : S_{t_i} \leq \min_{t_i \in Desc(e,T)} S_{t_i}. \tag{7}$$

4 Experimental Setup

4.1 Baselines

Since supervised learning with a deep learning approach in entity summarization tasks substantially outperforms unsupervised learning-based entity summarization tasks, we only consider the following methods as the baselines:

1. **ESA** [28] exploits graph embedding to encode triples of entity descriptions. Triple scores are calculated by leveraging normalized attention weights based on output vectors of BiLSTM computation.
2. **AutoSUM** [29] improves the ESA model by incorporating multi-user preference simulations such as entity and user phase attention.
3. **NEST** [12] leverages a KG encoder that represents structural and textual representations from KGs, employing joint learning from salience and diversified summary learning to produce the entity summary.
4. **DeepLENS** [15] uses textual semantics for triple encoding, and employs a deep learning model (such as BiLSTM) and MLP to generate scores for triples of entity descriptions.
5. **GATES** [8] computes triple scores using a combination of information from textual and structural representations generated by GNNs. Additionally, ensemble learning is used to improve triple-scoring performance.
6. **ESCS** [4] exploits description complementary and salience learning to score components of the input knowledge graphs. It then employs joint learning to calculate triple scores.

We used the DeepLENS, AutoSUM, and ESA experimental results presented in [8] as the code for these evaluation is open and the evaluation can be replicated. However, we could not use the NDCG approach to compare our model to NEST and ESCS due to unavailable codes.

4.2 Datasets

We used two types of benchmark datasets, including the ESBM (version 1.2)[1] and FACES. The ESBM comprises 125 and 50 entities which are from the DBpedia and LinkedMDB datasets, respectively. From the FACES dataset, we utilize 50 entities that are from the DBpedia dataset. The DBpedia dataset in ESBM separates the entities into five classes: agent, event, location, species, and work. The LinkedMDB dataset contains two distinct categories: films and persons. Each entity in every dataset is described by at least 20 triples. Additionally, FACES contains at least four manually generated entity summaries, while the ESBM provides six manually constructed ground truth summaries for each entity.

4.3 Experimental Settings

The ESLM is initialized by incorporating pre-trained LMs to leverage the knowledge already acquired. Specifically, we utilized *bert-base-uncased*[2] for integrating the BERT model and *ernie-2.0-en*[3] for ERNIE. Both models comprise 12 layers, 12 self-attention heads, and 768 hidden layers H. For the T5 model integration within ESLM, we employed the *t5-base*' LM[4]. The fine-tuning process was implemented on ESLM with varied learning rates $\in \{1 - 5 \times 10^{-5}\}$. Furthermore, we employed the binary cross-entropy (BCE) loss function as the primary criterion for training. The BCE loss is particularly well-suited for binary classification tasks, which aligns with the nature of our entity summarization problem where the model needs to predict the relevance of each entity triple within a given context. During training, the AdamW optimizer was employed alongside BCE. Throughout our evaluation, we used a five-fold cross-validation. In subsequent experiments, we used the ComplEx [26] method generated by the DGLKE framework[5] to enrich a PLM with KGE.

We conducted a statistical significance test to check whether the findings obtained by our models were significantly different from those of SOTA methods. In particular, we utilized the Wilcoxon signed ranked test with a 95% significance level. All experiments were conducted on a 64-core AMD EPYC 7713 CPU (2.0 GHz) with 1024 GB of RAM, running on Debian with CUDA using 2 NVIDIA GeForce RTX A5000 24 GB GPUs.

5 Results and Analysis

5.1 Comparison with State-of-the-Art Approaches

The baselines ESCS, GATES, DeepLENS, AutoSUM, NEST, and ESA are together referred to as SOTA. Table 1 shows average F-measure scores for differ-

[1] https://github.com/nju-websoft/ESBM/tree/master/v1.2.
[2] https://huggingface.co/bert-base-uncased.
[3] https://huggingface.co/nghuyong/ernie-2.0-base-en.
[4] https://huggingface.co/t5-base.
[5] https://github.com/awslabs/dgl-ke.

ent entity summarization models across three datasets: DBpedia, LinkedMDB, and FACES. These models include baseline models and our approach (ESLM) model, evaluated at the cut-offs $k = 5$ and $k = 10$, representing the top-5 and top-10 retrieved sets of triples, respectively. A clear pattern emerges: models generally improve their scores as the k value increases. This suggests a greater capacity to capture relevant triples of entities within a larger retrieval window. Notably, the ESLM model, which employs a combination of BERT, ERNIE, and T5 LMs determined to be suitable for entity summarization through an ablation study (see Sect. 5.3), outperforms others in DBpedia and FACES at both $k = 5$ and $k = 10$, indicating its superior summarization performance on these datasets.

The lower F-measure scores of ESLM model on the LinkedMDB dataset, specifically at $k = 5$ and $k = 10$, indicate that it may not be as well calibrated to the specific domain of the dataset as the ESCS and DeepLENS models. Although ESLM's performance shows improvement when considering a broader range of top predictions, it still falls behind models such as AutoSUM, GATES, and ESCS. The results underscore the need for ESLM to optimize its feature extraction and integrate KGE more effectively to enhance its summarization quality, indicating room for further refinement of the model's approach.

Table 1. Average F-measure score based on our model testing via five-fold cross-validation processes to all entities of the benchmark.

	DBpedia		LinkedMDB		FACES	
	k = 5	k = 10	k = 5	k = 10	k = 5	k = 10
ESA	0.332	0.532	0.353	0.435	0.153	0.261
NEST	0.354	0.540	0.332	0.465	0.272	0.346
AutoSUM	0.372	0.555	0.430	0.520	0.241	0.316
DeepLENS	0.404	0.575	0.469	0.489	0.130	0.248
GATES	0.423	0.574	0.437	**0.535**	0.254	0.324
ESCS	0.415	0.582	**0.494**	0.512	-	-
ESLM	**0.427**	**0.591**	0.467	0.498	**0.301**	**0.369**

NDCG scores for ESLM and the baselines across DBpedia, LinkedMDB, and FACES datasets are presented in Table 2. NDCG is a performance metric for the quality of ranked order outputs, with higher scores indicating that a model is effectively ranking highly relevant triples of entities at the top of the list. Our model ESLM clearly outperforms other methods in all datasets at both $k = 5$ and $k = 10$, with its peak score on DBpedia at $k = 10$ being 0.913. In comparison, while DeepLENS, GATES, and AutoSUM demonstrate strong performance with scores over 0.800 in several instances, they fall short of ESLM's consistency across datasets and cutoff values. The improvement from $k = 5$ to $k = 10$ for all models suggests that they are more adept at providing quality entity summarizations

when more results are included. In particular, ESLM's superior NDCG scores at both levels indicate its robust capacity to synthesize and rank entity information effectively using LM-driven approaches. This performance implies that ESLM has a significant advantage in tasks requiring nuanced discernment of entity triple relevance, especially in larger result sets.

Table 2. Average NDCG score based on our model testing via five-fold cross-validation processes to all entities of the benchmark.

	DBpedia		LinkedMDB		FACES	
	k = 5	k = 10	k = 5	k = 10	k = 5	k = 10
ESA	0.755	0.846	0.737	0.799	0.601	0.707
AutoSUM	0.797	0.882	0.809	0.856	0.693	0.768
DeepLENS	0.825	0.905	0.855	0.888	0.585	0.715
GATES	0.798	0.893	0.804	0.881	0.697	0.759
ESLM	**0.850**	**0.913**	**0.868**	**0.893**	**0.735**	**0.793**

We also ran a statistical test to see whether our method significantly outperforms the current SOTA benchmarks. We regard a p-value less or equal to 0.05 as significant. As shown in Table 3, ESLM performs significantly better than the compared models in most cases. In particular, the results show that ESLM outperforms the existing SOTA significantly in at least one experimental setting on the DBpedia dataset from ESBM and the FACES datasets. Additionally, ESLM significantly outperforms ESA, AutoSUM, and GATES across all evaluated $k = 5$ in the LinkedMDB dataset. The provided p-values confirm the statistical significance of our findings, with ESLM demonstrating a clear advantage over ESA, AutoSUM, DeepLENS, and GATES in almost all settings.

Table 3. Comparison of ESLM and SOTA using a Wilcoxon-rank test on the F-measure scores. The leftmost column shows the approaches ESLM was compared with. The values in the table are p-values.

	DBpedia		LinkedMDB		FACES	
	k = 5	k = 10	k = 5	k = 10	k = 5	k = 10
ESA	0.000	0.000	0.000	0.004	0.000	0.000
AutoSUM	0.000	0.000	0.028	-	0.003	0.004
DeepLENS	0.013	0.015	-	0.672	0.000	0.000
GATES	0.505	0.002	0.000	-	0.016	0.001

5.2 Example Findings

This section describes two examples found in our on test results. Due to space constraints, we chose the strongest baseline, GATES, and our model, ESLM, as well as a form of ground truth. In Fig. 2, we see a direct comparison between entity summaries produced by the ESLM and GATES models for a given entity, `Ludwigsburg University`. The output of ESLM is well aligned with the ground truth summary, capturing a subset of triples that directly correspond to the essential information about the entity. This is reflected in its higher F-measure score of 0.666, surpassing GATES' score of 0.600. A key observation is that GATES omits crucial `label` and `type` information in its summary, which indicates a gap in capturing and presenting significant details. ESLM's ability to include these important triples results in a more accurate and comprehensive summary enhances the model's utility in applications that depend on detailed entity representations.

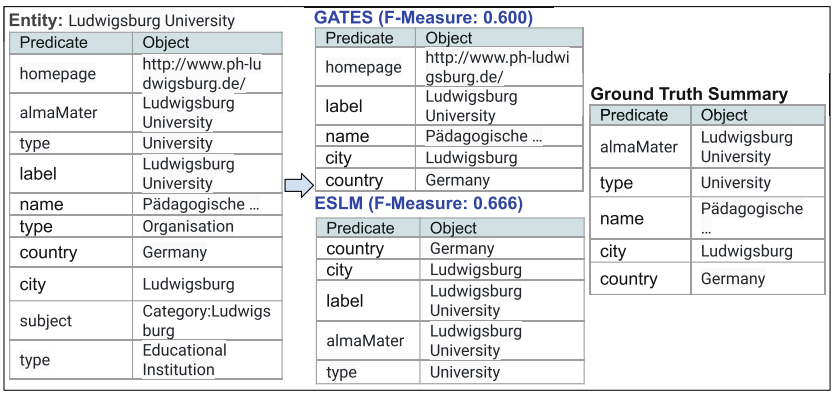

Fig. 2. Effectiveness of ESLM compared to GATES in use case one.

Figure 3 shows another example where ESLM outperforms GATES concerning entity summary results. Here, ESLM's F-measure improves from 0.533 to 0.733 by providing a wide range of information on the target entity, whereas the GATES lacks diversity in the information it provides, as it tends to repeat information about places.

According to both use cases, ESLM demonstrates that the use of LMs for entity summarization tasks relatively improves the quality of the summaries.

5.3 Ablation Study

In a preliminary assessment of the ESLM model, we compared the performance of pre-trained LMs such as BERT, ERNIE, and T5 on the DBpedia, LinkedMDB, and FACES datasets without the enhancement of KGE. The F-measure served

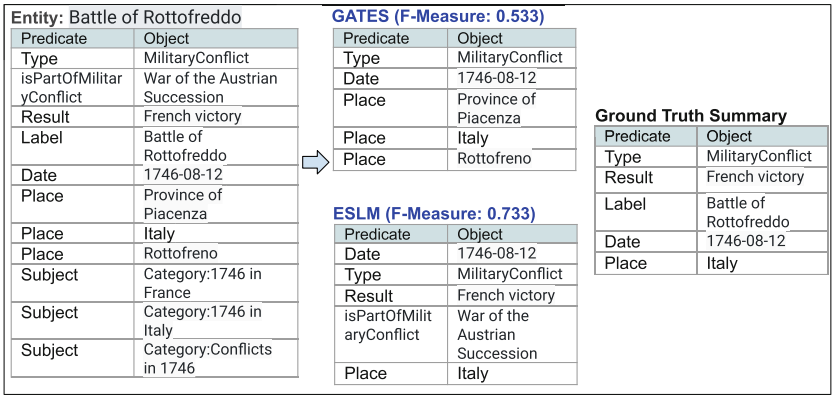

Fig. 3. Effectiveness of ESLM compared to GATES in use case two.

as the basis for this comparison. As evidenced by the results in Table 4, ERNIE demonstrates consistent superiority over BERT and T5 for $k = 5$ across all datasets. When examining the results for $k = 10$, T5 marginally outperforms BERT on DBpedia, whereas BERT maintains a slight advantage over T5 on LinkedMDB.

The second ablation study's results are shown in Table 4. These results indicate that KGEs contribute positively to the F-measure performance of the BERT, ERNIE, and T5 LMs on three distinct datasets. On the DBpedia and LinkedMDB datasets, BERT and T5's performances are notably boosted by KGE, with T5 showing the most significant enhancement in DBpedia for $k = 5$, increasing its score from 0.410 to 0.427. ERNIE exhibits smaller gains with the addition of KGE, suggesting its pre-existing pre-training might already include some of the relational knowledge that KGEs introduce. In contrast, on the FACES dataset, T5 especially gains at $k = 10$. These results suggest that the degree of alignment between a model's pre-training and a dataset can be a crucial determinant of the effectiveness of integrating KGEs. The consistent increase in F-measure scores from $k = 5$ to $k = 10$ suggests that KGEs provide supplementary information that can improve the models' capability to effectively rank relevant information.

We also used NDCG scores to examine the BERT, ERNIE, and T5 models on the ESLM model, evaluating their ranking effectiveness both with and without KGE, as shown in Table 5. BERT exhibits marginal improvements with KGE integration, with the NDCG score slightly increasing across the DBpedia and LinkedMDB datasets for both the top-5 and top-10 results but showing minimal change on the FACES dataset. ERNIE's performance with KGE is mixed. It experiences a slight decrease in the DBpedia dataset but improvements in LinkedMDB and FACES, particularly at $k = 10$ in LinkedMDB. T5 demonstrates the most significant gains from KGEs across all datasets, especially at $k = 10$ in LinkedMDB, suggesting that T5's architecture effectively utilizes the

Table 4. Highest F-measure performance of BERT, ERNIE, and T5 on ESLM

Models	DBpedia		LinkedMDB		FACES	
	k = 5	k = 10	k = 5	k = 10	k = 5	k = 10
BERT	0.411	0.574	0.444	0.494	0.286	0.355
BERT + KGE	0.417	0.586	0.445	0.482	**0.301**	0.347
ERNIE	0.421	0.583	0.448	0.482	0.292	0.348
ERNIE + KGE	0.423	0.586	**0.467**	0.494	0.295	**0.369**
T5	0.410	0.584	0.442	0.486	0.287	0.352
T5 + KGE	**0.427**	**0.591**	0.455	**0.498**	0.300	0.361

additional relational knowledge provided by KGEs to enhance ranking accuracy. The consistently higher scores at $k = 10$ across all models imply that the models perform better when evaluating a larger set of predictions, which is crucial for tasks involving entity summarization where multiple correct answers are possible.

Table 5. Highest NDCG scores of BERT, ERNIE, and T5 on ESLM

Models	DBpedia		LinkedMDB		FACES	
	k = 5	k = 10	k = 5	k = 10	k = 5	k = 10
BERT	0.841	0.904	0.836	0.848	0.736	**0.797**
BERT + KGE	0.842	0.908	0.835	0.840	0.733	0.796
ERNIE	0.841	0.910	0.831	0.837	0.740	0.788
ERNIE + KGE	0.838	0.909	0.867	0.858	**0.744**	0.795
T5	**0.852**	0.908	0.862	0.869	0.742	0.785
T5 + KGE	0.850	**0.913**	**0.868**	**0.893**	0.736	0.790

5.4 Computational Requirements and Efficiency of ESLM

Table 6 outlines the training times and processing efficiencies of among ESLM models with and without KGE, trained on the DBpedia, LinkedMDB, and FACES datasets over 50 epochs using 1 GPU. The integration of contextual LMs and KGEs slightly increases training times across all models, suggesting a modest rise in computational requirements. The T5 model is the most time-consuming. All configurations process a comparable number of entities per second, with a negligible increase when KGE is included. This observation is important as it implies that the addition of KGEs enhances the performance of supervised entity summarization approaches without increasing their computational needs.

Table 6. Comparative Analysis of Training Times and Entity Processing Efficiency among ESLM models. All times are seconds, and the total number of epochs is 50.

Models	Topk	Input	Output	Training Time		Prediction Time
		Triples	Triples	Total	Mean	Single Triples
BERT	5	4436	750	329.56	6.59	0.060
BERT + KGE	5	4436	750	404.50	8.09	0.060
ERNIE	5	4436	750	327.39	6.55	0.067
ERNIE + KGE	5	4436	750	328.88	6.58	0.070
T5	5	4436	750	402.96	8.06	0.071
T5 + KGE	5	4436	750	411.86	8.24	0.072
BERT	10	4436	1500	333.62	6.67	0.060
BERT + KGE	10	4436	1500	329.91	6.60	0.059
ERNIE	10	4436	1500	333.32	6.67	0.069
ERNIE + KGE	10	4436	1500	329.24	6.58	0.069
T5	10	4436	1500	403.83	8.07	0.070
T5 + KGE	10	4436	1500	413.38	8.27	0.073
BERT	5	2148	125	184.06	3.68	0.123
BERT + KGE	5	2148	125	185.14	3.70	0.125
ERNIE	5	2148	125	184.85	3.70	0.144
ERNIE + KGE	5	2148	125	185.82	3.72	0.144
T5	5	2148	125	188.05	3.76	0.151
T5 + KGE	5	2148	125	189.40	3.79	0.154
BERT	10	2148	250	185.65	3.71	0.173
BERT + KGE	10	2148	250	185.71	3.71	0.123
ERNIE	10	2148	250	185.92	3.72	0.157
ERNIE + KGE	10	2148	250	185.82	3.72	0.145
T5	10	2148	250	189.70	3.79	0.153
T5 + KGE	10	2148	250	188.20	3.76	0.155
BERT	5	2152	125	186.47	3.73	0.122
BERT + KGE	5	2152	125	186.22	3.72	0.126
ERNIE	5	2152	125	186.32	3.73	0.142
ERNIE + KGE	5	2152	125	186.37	3.73	0.154
T5	5	2152	125	188.09	3.76	0.171
T5 + KGE	5	2152	125	189.31	3.79	0.154
BERT	10	2152	250	188.55	3.77	0.124
BERT + KGE	10	2152	250	186.85	3.74	0.126
ERNIE	10	2152	250	187.27	3.75	0.147
ERNIE + KGE	10	2152	250	187.02	3.74	0.146
T5	10	2152	250	191.41	3.83	0.156
T5 + KGE	10	2152	250	190.75	3.82	0.157

6 Conclusion and Future Work

In this study, we introduced ESLM, an entity summarization method leveraging LMs enhanced with KGEs. Our analysis showed that ERNIE-based implementations of ESLM outperform BERT-based approaches, with further improvements when these models are enriched with KGEs, particularly in the T5 model. Our results also suggest that ESLM achieves significantly better results than the cur-

rent SOTA methods, as evidenced by ESLM superior performance on benchmark datasets such as DBpedia and FACES. Despite these advancements, we recognize a limitation in the scale of current gold standard datasets, such as ESBM and FACES. This highlights a broader issue in the field's lack of comprehensive benchmarking datasets for entity summarization. To address this challenge, we plan to develop an extensive silver dataset to support the creation of robust and reliable entity summarization models. Additionally, we aim to explore the integration of ESLM with graph neural networks, potentially enhancing our model's capabilities further.

Acknowledgements. This work has been supported by the Ministry for Economic Affairs, Innovation, Digitalisation and Energy of North Rhine-Westphalia (MWIDE NRW) within the project Climate bOWL (grant no. 005-2111-0020), the German Federal Ministry of Education and Research (BMBF) within the projects KIAM (grant no. 02L19C115), COLIDE (grant no. 01I521005D), the European Union's Horizon Europe research and innovation programme (grant no. 101070305), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824, and Mora Scholarship from the Ministry of Religious Affairs, Republic of Indonesia.

References

1. Ali, M., Saleem, M., Ngomo, A.C.N.: Unsupervised relation extraction using sentence encoding. In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, pp. 136–140. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80418-3_25
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003). http://jmlr.org/papers/v3/bengio03a.html
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
4. Chen, L., et al.: Entity summarization via exploiting description complementarity and salience. IEEE Trans. Neural Netw. Learn. Syst. (2022)
5. Cheng, G., Tran, T., Qu, Y.: RELIN: relatedness and informativeness-based centrality for entity summarization. In: Aroyo, L., et al. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 114–129. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_8
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
7. Ermilov, T., Moussallem, D., Usbeck, R., Ngomo, A.C.N.: Genesis: a generic RDF data access interface. In: Proceedings of the International Conference on Web Intelligence, pp. 125–131. WI 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3106426.3106514
8. Firmansyah, A.F., Moussallem, D., Ngomo, A.N.: GATES: using graph attention networks for entity summarization. In: Gentile, A.L., Gonçalves, R. (eds.) K-CAP 2021: Knowledge Capture Conference, Virtual Event, USA, December 2-3, 2021, pp. 73–80. ACM (2021). https://doi.org/10.1145/3460210.3493574

9. Gunaratna, K., Thirunarayan, K., Sheth, A.: Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 116–122. AAAI Press (2015)
10. Hasibi, F., Balog, K., Bratsberg, S.E.: Dynamic factual summaries for entity cards. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 773–782. SIGIR 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3077136.3080810
11. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: compressing text classification models. CoRR **abs/1612.03651** (2016), http://arxiv.org/abs/1612.03651
12. Li, J., et al.: Neural entity summarization with joint encoding and weak supervision. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 1644–1650. ijcai.org (2020). https://doi.org/10.24963/ijcai.2020/228
13. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: ESBM: an entity summarization BenchMark. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 548–564. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_32
14. Liu, Q., Cheng, G., Gunaratna, K., Qu, Y.: Entity summarization: state of the art and future challenges. J. Web Semant. **69**, 100647 (2021). https://doi.org/10.1016/j.websem.2021.100647
15. Liu, Q., Cheng, G., Qu, Y.: Deeplens: deep learning for entity summarization. CoRR **abs/2003.03736** (2020). https://arxiv.org/abs/2003.03736
16. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: Barzilay, R., Kan, M.Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1789–1798. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1164, https://aclanthology.org/P17-1164
17. Liu, Y.: Fine-tune BERT for extractive summarization. CoRR **abs/1903.10318** (2019). http://arxiv.org/abs/1903.10318
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013). http://arxiv.org/abs/1301.3781
19. Patil, P., Rao, C., Reddy, G., Ram, R., Meena, S.M.: Extractive text summarization using BERT. In: Gunjan, V.K., Zurada, J.M. (eds.) Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. LNNS, vol. 237, pp. 741–747. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-6407-6_63
20. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1162
21. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1) (2020)
22. Sacenti, J.A., Fileto, R., Willrich, R.: Knowledge graph summarization impacts on movie recommendations. J. Intell. Inf. Syst. **58**(1), 43–66 (2022)

23. Sun, Y., et al.: ERNIE 2.0: a continual pre-training framework for language understanding. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 8968–8975. AAAI Press (2020). https://ojs.aaai.org/index.php/AAAI/article/view/6428
24. Sydow, M., Pikula, M., Schenkel, R.: DIVERSUM: towards diversified summarisation of entities in knowledge graphs. In: Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA, pp. 221–226. IEEE Computer Society (2010). https://doi.org/10.1109/ICDEW.2010.5452707
25. Thalhammer, A., Lasierra, N., Rettinger, A.: LinkSUM: using link analysis to summarize entity data. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 244–261. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_14
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
27. Wei, D., Gao, S., Liu, Y., Liu, Z., Hang, L.: MPSUM: entity summarization with predicate-based matching. CoRR **abs/2005.11992** (2020). https://arxiv.org/abs/2005.11992
28. Wei, D., Liu, Y.: ESA: entity summarization with attention. CoRR **abs/1905.10625** (2019). http://arxiv.org/abs/1905.10625
29. Wei, D., et al.: AutoSUM: automating feature extraction and multi-user preference simulation for entity summarization. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12085, pp. 580–592. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47436-2_44
30. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. CoRR **abs/1909.03193** (2019). http://arxiv.org/abs/1909.03193
31. Zaib, M., Tran, D.H., Sagar, S., Mahmood, A., Zhang, W.E., Sheng, Q.Z.: BERT-CoQAC: BERT-based conversational question answering in context. In: Ning, L., Chau, V., Lau, F. (eds.) PAAP 2020. CCIS, vol. 1362, pp. 47–57. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0010-4_5
32. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 1441–1451. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/p19-1139