Revisiting ESLM: Strengths, Weaknesses and Further Experiments

Seminar Paper

by

Simon Schmitt

Degree Course: Information Systems M.Sc.
Matriculation Number: 2279089

Institute of Applied Informatics and Formal Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Dr. Genet Asefa Gesese
Second Advisor:	Mary Ann Tan, M.Sc.
Supervisor:	Prof. Dr. Harald Sack
Submitted:	March 19, 2025

Contents

1 Introduction

This report discusses the paper "ESLM: Improving Entity Summarization by Leveraging Language Models" [6]. This chapter covers entity summarization basics, existing approaches, and research questions. Chapter 2 provides an in-depth explanation of the ESLM architecture, while Chapter 3 discusses the results from the original paper. Chapter 4 provides the strengths, limitations, and future research directions, followed by Chapter 5, which describes additional experiments to enhance the ESLM architecture.

1.1 Entity Summarization

The aim of entity summarization is to extract a subset of the most relevant triples from the potentially large number of triples associated with an entity of interest. Entity summarization can be applied in all kinds of practical areas such as web search, RDF browsers or recommender systems where information from massive graph structures has to be condensed [6]. Given an entity of interest e, the entity description $Desc(e, G)$ contains all triples t associated with e in the knowledge graph G. Entity summarization approaches always apply a triple scoring algorithm that assigns triple scores S_t representing the relative importance of each triple. These triple scores are then used to extract the k triples with the highest relevance that end up in the entity summary $ES(e)$. The formal condition that the entity summarization has to meet can be seen below: It says that the triple t_i from the entity summary $ES(e)$ with the lowest triple score S_{t_i} has to have a triple score greater than or equal to all triples t' that did not make it into the entity summary.

$$\forall t' \in Desc(e, G) \setminus ES(e) : S_{t'} \leq \min_{\forall t_i \in ES(e)} S_{t_i} \tag{1}$$

1.2 Existing Approaches

Today, deep learning-based approaches such as GATES [5] or ESA [16] deliver the best performance for entity summarization. What they all have in common is that some kind of neural network architecture is used to generate representations for all triples. The triple scores are then determined on the basis of these representations. ESA was one of the earlier approaches and laid the foundation for many of the following research papers. In this model, the predicate and object are encoded separately for each triple in the entity description and then concatenated. A word embedding (Neural probabilistic model [2]) is used for the predicate and a knowledge graph embedding (TransE [3]) for the object. The triple representation vectors are then processed by a BiLSTM network [7] in which each triple is encoded on the basis of information about previous (forward LSTM) and subsequent (backward LSTM) triples. Finally, there is an attention layer [1] that as-

signs attention weights to the triples. The output of the model is compared with entity summaries generated by human experts and the model parameters are updated via supervised learning. ESA provides significantly better results than unsupervised approaches used previously, but also has some limitations.
First of all, a static word embedding is used for encoding the predicate, which means that the same word is always encoded in the same way regardless of the context. This can lead to important context information being lost. In addition, the separate encoding of predicate and object without [SEP] tokens makes it difficult to recognize dependencies between them later, as the model no longer knows where the individual components begin or end. There is also no dedicated layer for a more sophisticated integration of textual and knowledge graph representations. The representation vectors are simply concatenated without any information weighting. And finally, the BiLSTM network limits the performance of the model because the triples always have to be processed sequentially.

1.3 Research Questions

Since existing approaches such as ESA have limitations, the authors of the paper discussed here build their work around the hypothesis that the use of language models can improve entity summarization. More specifically, they examine three research questions.
The first research question deals with which configurations of the proposed ESLM architecture deliver the best results. Based on this, the second research question aims to find out whether the ESLM architecture can outperform existing state of the art (SOTA) approaches such as ESA. The final question is how the individual model configurations perform in terms of computational effort and processing efficiency.

2 ESLM Architecture

Figure 1, which shows the ESLM architecture, was created specifically for this report and attempts to show the processes more clearly than the illustration in the original paper.

2.1 Preprocessing and Tokenization

The triples to be processed from the entity description initially pass through a path for the textual representation and a path for graph structural representation. In the first step of the textual representation, preprocessing and tokenization, the individual triple components are separated by [SEP] tokens and formatted into one string per triple. During the subsequent tokenization, this string is transformed into a sequence of smaller, machine-readable units. It is noticeable that the ESLM architecture overcomes a limitation of ESA here, since triples are always processed as a whole and thus dependencies between the individual components can be better recognized.

2.2 Language Model Encoder

In the second step of the textual representation, the sequence of tokens for each triple is processed by a language model encoder. The language model encoder generates contextual embeddings for each token. This is made possible by the self-attention mechanism in the transformer blocks from which the language model encoder is built. Here, each token in the sequence is scored against the token that is currently being encoded. The score determines the extent to which the information of the respective token is taken into account when encoding the current token [15]. As a result, the embedding of each token contains context information about surrounding tokens and is therefore much more meaningful than the static word embeddings used in the ESA architecture and many other SOTA approaches. Since the training of large language models is very resource-intensive, the authors of this paper use three pre-trained publicly available encoder models, namely BERT [4], ERNIE 2.0 [13] and T5 [11]. In terms of their architecture and configuration, all three models are very similar (see Table 3 in appendix). Among other things, all three output the same hidden dimension of 768 for the contextual embeddings. The main difference lies in the type of learning tasks used during pre-training to inject knowledge (see Table 4 in appendix).

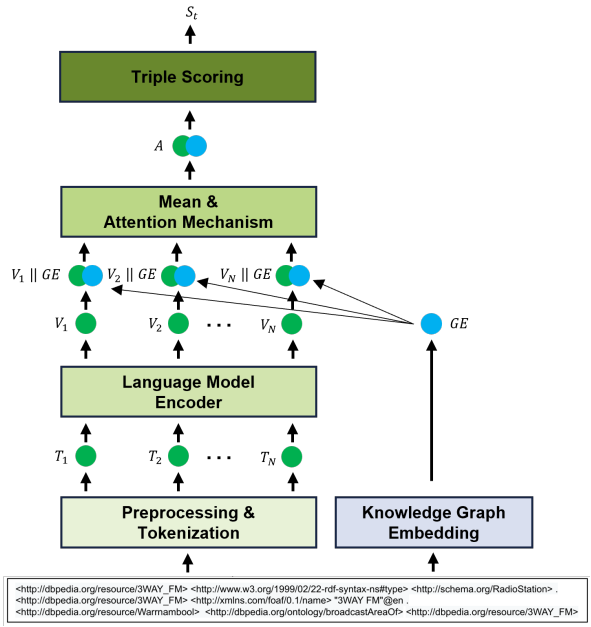

Figure 1: The ESLM model architecture

2.3 Knowledge Graph Embedding

The graph structural representation includes the creation of knowledge graph embeddings (KGE) for all three components of a triple. For this purpose, the authors of this paper use a technique called ComplEX [14]. Entities and relations are represented in a complex-

valued vector space (consisting of a real and an imaginary dimension), which makes this approach more expressive and computationally efficient than others. This results in embeddings of dimension 400 for subject, predicate and object which are then concatenated to form a vector of dimension 1200 for each triple.

2.4 Mean and Attention Mechanism

The next step is to combine textual and graph structural representation. To do this, the KGE of the triple is concatenated with each contextual token embedding of the corresponding triple sequence. In order to obtain a single representation vector per triple, the mean value is then computed over this sequence of vectors. The resulting vector V then passes through an attention layer [1] (see Formula 2). This allows the model to focus selectively on certain entries of the triple representation vector. In particular, this avoids a limitation of the ESA architecture, as there is now an explicit weighting and integration of textual and graph structural information.

$$A_{\text{weights}} = \text{Softmax}(VW_{\text{attn}} + b_{\text{attn}}), \quad A = A_{\text{weights}} \odot V \tag{2}$$

2.5 Triple Scoring

In the last step, the representation vectors for each triple are passed through a multilayer perceptron. The input layer has dimension 1968, followed by two hidden layers with dimension 512 and an output layer that outputs a single value for each triple. Finally, a softmax activation function is applied to these values and the result are the final triple scores S_t.

$$S_t = \sigma(\text{MLP}(A)) \tag{3}$$

3 ESLM Experiment

To evaluate the performance of the ESLM architecture, a benchmark with existing deep learning-based approaches is performed in the ESLM paper.

3.1 Datasets

Three datasets are used for the benchmark (see Table 5 in appendix). Each dataset consists of a set of entities and the corresponding triples from the knowledge graph. For each entity, several human-generated gold solutions are given for the corresponding entity summarization. The DBpedia [10] and FACES [8] datasets are based on a structured

representation of Wikipedia and contain different types of entities. LinkedMDB [10], on the other hand, is based on the IMDb film database and specifically contains entities from the film domain.

3.2 Training

As part of the benchmark, the different model configurations of ESLM are evaluated on the three datasets. The model configuration is determined by the language model encoder used (BERT, ERNIE 2.0 or T5). In addition, the omission of KGE (see Chapter 2.3) is tested in each case. The model configurations are evaluated on all three datasets for both top 5 and top 10 entity summarization. In the evaluation for each model configuration, five different model instances are trained using 5-fold cross-validation and each tested on a different part of the dataset. The training parameters used during training can be found in Table 6 in the appendix. The evaluation results of the SOTA approaches are taken from the respective research papers.

3.3 Evaluation Metrics

The F-measure and the Normalized Discounted Cumulative Gain (NDCG) are used as metrics for the evaluation. The F-measure combines precision and recall into one metric and prefers models with a balanced distribution of false positives and false negatives. The NDCG measures the quality of rankings. Here, a discounted gain is calculated for the ranking predicted by ESLM based on the triple scores and for the ideal ranking based on the gold solutions, and the ratio is then taken. As a result of discounting, higher positions have a greater influence on the value of the metric.

3.4 Results

With regard to the first research question, namely which model configuration of ESLM performs best, the benchmark results show that the use of T5 as encoder with KGE leads to the best results. Table 5 in the original paper shows that this configuration achieves the highest values for the NDCG on average across all experiment settings. A similar picture emerges for the F-measure (see Table 4 in original paper). It can be seen that the scores for ERNIE 2.0 and T5 increase when KGE is added, thereby adding relevant information. Furthermore, the results show that for $k = 10$ the performance is generally much better, which indicates that the models find a larger retrieval window easier.
The second research question was whether ESLM can outperform SOTA approaches. For this purpose, the best performing ESLM configuration is used for each experiment setting and compared with the existing approaches. Table 1 shows that ESLM achieves better F-measure scores than all other SOTA approaches on average across all experiment settings.

To verify the results, a Wilcoxon-rank test on the F-measure is carried out (see Table 3 in original paper). This shows that the better performance of ESLM is also significant in most cases. However, in some experiment settings other approaches are ahead and evaluation results for some of the SOTA approaches are missing. For the NDCG (see Table 2 in original paper), ESLM performs best in every single experiment setting.

Table 1: F-measure benchmark scores for ESLM and SOTA approaches

Models	DBpedia		LinkedMDB		FACES		Mean
	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	0.344
NEST	0.354	0.540	0.332	0.465	0.272	0.346	0.385
AutoSUM	0.372	0.555	0.430	0.520	0.241	0.316	0.406
DeepLENS	0.404	0.575	0.469	0.489	0.130	0.248	0.386
GATES	0.423	0.574	0.437	**0.535**	0.254	0.324	0.425
ESCS	0.415	0.582	0.494	0.512	-	-	-
ESLM	**0.427**	**0.591**	0.467	0.498	**0.301**	**0.369**	**0.442**

The final research question concerns the extent to which the individual model configurations of ESLM differ from each other in terms of training effort and inference time. To this end, the authors of this paper take and evaluate the time required for training and inference in each case. The results can be found in Table 7 in the appendix. The evaluation in Table 8 in the appendix was specifically created for this report and shows that model configurations with T5 are on average the most computationally intensive in training and also have the highest inference time. It is also clear that the integration of KGE increases the training and inference time slightly but to a negligible extent. This is important as it shows that KGE add value to the prediction quality but at the same time do not significantly increase the computational effort. Apart from that, it would have been interesting to look at the training and inference times for a model without a language model encoder (e.g. ESA). This would have clarified to what extent the encoder in the ESLM architecture increases the computational effort.

4 Discussion

This section evaluates the contributions of the ESLM paper and discusses its strengths and weaknesses. Possible future research directions are also outlined.

4.1 Strengths

As already mentioned in the description of the architecture in Chapter 2, ESLM overcomes many of the conceptual weaknesses of earlier approaches such as ESA. Among other things, the use of a language model encoder for the textual representation allows more contextual information about the original triple to be preserved. ESLM also performs very well in a benchmark comparison with SOTA approaches. It outperforms the other approaches on average for both metrics. The results can also be considered robust due to the evaluation on several experiment settings, the use of cross-validation and the verification by a statistical test.

4.2 Weaknesses and Further Research Directions

Although they do not necessarily invalidate the content of the paper, there are some methodological inaccuracies and errors that are worth mentioning. First of all, the original paper gives a different value for the KGE dimension used than was actually used for the experiments. In addition, the formula for the entity summarization (see Chapter 3.4 of original paper) is incorrect, as two sub-elements are mixed up. Also, in the table of computational requirements (see Chapter 3.4 of this report), the values for the number of output triples are incorrect for the various model configurations and it is not clear to which data set the results relate in each case. The most significant issue is certainly that the authors cited the wrong paper for the encoder model ERNIE. The reference is to ERNIE [17], which features an innovative approach for integrating knowledge graphs into pre-training. However, the experiments actually used ERNIE [12], which only happens to share the same name but does not employ the mentioned approach in pre-training. As a result, parts of the motivation and evaluation are misleading. The authors have acknowledged the points raised in email correspondence and will correct them in a revised version.

Although the overall benchmark results are impressive, it should also be noted that for ESCS, one of the most promising SOTA approaches, the results are not complete. In addition, one may question whether the comparison is not biased by the prior ablation study and the use of the best model configuration of ESLM for each individual experiment setting.

One issue that affects the research field of entity summarization in general is the lack of large and diverse benchmark datasets. The datasets used in this paper are all very small, well curated and cover too few classes. As a result, the evaluation results are only partially meaningful. A possible solution could be an approach called Wiki Entity Summarization (WikES) [9]. It automatically generates large-scale labeled benchmark datasets for entity summarization by matching Wikidata triples with Wikipedia references.

One step that is completely missing from the paper is a quantitative error analysis that

takes a look at the characteristics of the entities for which ESLM does not perform well. This could provide valuable insights for potential further developments and modifications. Therefore, a quantitative error analysis is carried out in Chapter 5.2 of this report.
A potential drawback of the ESLM architecture is that triples are processed and scored completely independently of one another, unlike most SOTA approaches. In particular, there is no cross-triple attention mechanism. However, scenarios are conceivable where the relevance of a triple for entity summarization might depend on the presence of other triples in the entity description (see Figure 2 in appendix). Therefore, in Chapter 5.3 of this report, the possibility of extending ESLM with a second-level encoder to introduce a mechanism for cross-triple attention is explored.
Finally, ESLM does not receive any contextual information about the entity summarization to be created. Knowledge about the class to which the entity belongs or the desired properties of the summary could improve the results. Accordingly, in Chapter 5.4 of this report, a prompting approach is tested to provide ESLM with more information.

5 Further Experiments

Based on the previous discussion, several modifications of the ESLM architecture are experimented with in this chapter.

5.1 Error Analysis

The experiments are preceded by a quantitative error analysis to evaluate for which types of entities ESLM does not perform well. For this purpose, the model configuration with T5 as encoder and KGE is used, as this performs best overall (see Chapter 3.4). The model configuration is trained as described above. In the test phase, the 30 percent entities for which ESLM achieves the lowest F-measure scores are extracted individually for each experiment setting. The first question that arises is whether the class of the entities has an influence on how well ESLM handles the entity summarization. For this, each entity is assigned its corresponding class from the original knowledge graph. Figure 3 in the appendix shows the class distribution for the entities extracted for the DBpedia benchmark dataset and top 5 summarization. While all five classes have the same frequency in the entire dataset, it can be clearly seen that the agent class (an entity that acts e.g. a person or organization) is the most frequent among those entities for which ESLM performs poorly. The event class (significant happenings e.g. historical or sport events), on the other hand, is significantly less represented.
The second question that arises is whether the node degree has an influence on the performance of ESLM. Table 9 in the appendix shows the median and mean value for the entire DBpedia benchmark dataset and for the subset of entities with the lowest F-measure score

for top 5 and top 10 summarization. Here it is clear that the node degree is significantly higher among those entities for which ESLM struggles. Similar patterns can also be observed for the LinkedMDB and FACES datasets.
All in all, it can be said that ESLM performs particularly poorly for entities from the agent class and for entities with a high node degree. However, it is not possible to say which of these factors is causally decisive on the basis of this evaluation, as entities of the agent class generally have a higher node degree than entities from other classes (see Table 10 in appendix).

5.2 Modified Information Fusion

As described in Chapter 2, in the ESLM architecture triples are processed as a sequence of tokens by the language model encoder and then the mean value is calculated from the resulting embeddings. This sequence of tokens has a fixed length of 40 tokens and if the sequence resulting from the tokenization is too short, it is padded with so-called padding tokens. These padding tokens have no semantic meaning and are masked during processing by the encoder, i.e. the other tokens cannot attend to these tokens. Despite this, the embeddings of the padding tokens, which have no actual meaning, are taken into account in the subsequent calculation of the mean. This experiment therefore tests if ignoring these tokens for the mean calculation leads to better entity summarizations. The results for the F-measure for this modified information fusion can be seen in Table 2. The best performing model configuration of ESLM (T5 with KGE) is used as the baseline. It can be seen that the modified information fusion approach beats ESLM on some experiment settings. However, ESLM performs slightly better on average over all experiment settings for both the F-measure and the NDCG (see Table 11 in appendix). Since the results are very close to each other, only evaluation on a larger benchmark dataset will be able to clarify whether this modificiation is beneficial or not.

Table 2: F-measure scores for the ESLM baseline (T5 + KGE) and its modifications

Approach	**DBpedia**		**LinkedMDB**		**FACES**		**Mean**
	$k = 5$	$k = 10$	$k = 5$	$k = 10$	$k = 5$	$k = 10$	
T5 + KGE	0.417	0.586	0.463	0.495	0.288	0.349	0.433
T5 + KGE + Modified Information Fusion	0.417	0.587	**0.465**	0.483	0.288	0.350	0.432
T5 + KGE + Second-level Encoder	0.403	0.567	0.409	0.504	0.230	0.303	0.403
T5 + KGE + Prompting	**0.424**	**0.590**	0.453	**0.505**	**0.294**	**0.353**	**0.437**

5.3 Second-level Encoder

As mentioned in Chapter 4.2, this experiment attempts to improve ESLM by integrating a mechanism for cross-triple attention. Unlike in many SOTA approaches, no BiLSTM

network is to be used here, as this has problems recognizing long range dependencies in a triple sequence and is also poorly scalable. Instead, a transformer encoder is inserted between the mean calculation and the attention mechanism. Here, however, not a sequence of tokens but a sequence of triples is processed. Projection layers are also added to adapt the dimensions of the input and output embeddings to the second-level encoder. T5 is used as the model for the second-level encoder, as it performed best in the ESLM ablation study. The comparison with the ESLM baseline shows that the second-level encoder approach only achieves a higher F-measure score for a single experiment setting. In all other experiment settings as well as in terms of NDCG, ESLM performs significantly better. One possible explanation for this could be that most of the triples in the entity description itself are only of low relevance. The cross-triple attention mechanism would therefore tend to distort the triple encodings and thus lead to less accurate triple scores.

5.4 Prompting

Another modification of the ESLM architecture discussed in Chapter 4.2 involves the inclusion of context information through prompting. This experiment specifically investigates whether additional knowledge about the name and class of the entity to be summarized can improve the results. For this purpose, the class is inserted together with the entity name in a context string, which is added at the beginning of each triple string (see example in Figure 4 in appendix). A look at the benchmark results shows that the prompting approach shows a strong performance for the F-measure and outperforms the ESLM baseline on average across all experiment settings. For the NDCG, it is a close race, with both approaches performing equally on average across all experiment settings.

5.5 Conclusion

The detailed analysis of the ESLM approach conducted in this report underlines that the use of contextual language models is promising for the task of entity summarization. ESLM eliminates many limitations of existing SOTA approaches and outperforms them in the benchmark study. However, as mentioned in Chapter 4, the original paper also has some methodological weaknesses, just as the ESLM architecture leaves room for further improvements, three of which were experimented with in Chapter 5. The modified information fusion and the second-level encoder are favorable for certain scenarios, but need further development and refinement to seriously compete with the ESLM baseline. The combination of ESLM with prompting already delivers results on a par with or even better than the ESLM baseline in its current form. Therefore, this report comes to the conclusion that the extension of ESLM with prompting components is the most promising way to achieve even better results for entity summarization in the future.

A Appendix

Table 3: Comparison of BERT [4], ERNIE 2.0 [13] and T5 [11]

	BERT	**ERNIE 2.0**	**T5**
Release	10/2018	03/2019	10/2019
Architecture	Encoder	Encoder	Encoder-Decoder
Variant	Base	Base	Base
Model Size	$\sim$ 110M	$\sim$ 110M	$\sim$ 220M/110M
# **Heads**	12	12	12
# **Transformer Blocks**	12	12	12
Hidden Dimension	768	768	768

Table 4: Comparison of BERT [4], ERNIE 2.0 [13] and T5 [11] regarding pre-training

	BERT	**ERNIE 2.0**	**T5**
Datasets	English Wikipedia, BooksCorpus	English Wikipedia, Discussion Data, ...	C4
Pre-Training Tasks	Masked Language Modeling (MLM), Next Sentence Prediction (NSP)	Continual Multi-Task-Learning (Knowledge Masking, Sentence Reordering, ...)	Denoising Objective (with Span Corruption)

Table 5: Comparison of DBpedia [10], LinkedMDB [10] and FACES [8] benchmark datasets

	DBpedia	**LinkedMDB**	**FACES**
Source	Wikipedia	IMDb	Wikipedia
# **Entities**	125	50	50
# **Classes**	5	2	17
# **Gold Solutions**	6	6	>4

Table 6: Training parameters used for the benchmark

Framework	PyTorch
Optimizer	AdamW
Learning Rate	1.5×10^{-5}
Scheduler	Linear decreasing learning rate (no warmup)
Epochs	10
Loss	Binary Cross-Entropy (BCE)

Table 7: Computational efficiency of ESLM model configurations for training and inference

Models	Topk	Input Triples	Output Triples	Training Time Total	Training Time Mean	Prediction Time Single Triples
DBpedia						
BERT	5	4436	625	329.56	6.59	0.060
BERT + KGE	5	4436	625	340.50	6.89	0.067
ERNIE	5	4436	625	327.39	6.55	0.067
ERNIE + KGE	5	4436	625	338.86	6.58	0.070
T5	5	4436	625	402.96	8.06	0.071
T5 + KGE	5	4436	625	411.86	8.24	0.072
BERT	10	4436	1250	333.62	6.67	0.059
BERT + KGE	10	4436	1250	329.91	6.60	0.059
ERNIE	10	4436	1250	330.32	6.67	0.069
ERNIE + KGE	10	4436	1250	329.24	6.58	0.069
T5	10	4436	1250	403.83	8.07	0.070
T5 + KGE	10	4436	1250	413.38	8.27	0.073
LinkedMDB						
BERT	5	2148	250	184.06	3.68	0.123
BERT + KGE	5	2148	250	185.14	3.70	0.125
ERNIE	5	2148	250	184.85	3.70	0.144
ERNIE + KGE	5	2148	250	185.14	3.70	0.144
T5	5	2148	250	188.05	3.76	0.151
T5 + KGE	5	2148	250	188.33	3.76	0.154
BERT	10	2148	500	185.65	3.71	0.125
BERT + KGE	10	2148	500	185.80	3.71	0.125
ERNIE	10	2148	500	185.92	3.72	0.145
ERNIE + KGE	10	2148	500	185.82	3.72	0.145
T5	10	2148	500	188.20	3.74	0.146
T5 + KGE	10	2148	500	188.20	3.76	0.155
FACES						
BERT	5	2152	250	186.47	3.73	0.122
BERT + KGE	5	2152	250	186.22	3.73	0.123
ERNIE	5	2152	250	187.20	3.73	0.142
ERNIE + KGE	5	2152	250	187.27	3.75	0.143
T5	5	2152	250	188.09	3.79	0.146
T5 + KGE	5	2152	250	190.75	3.82	0.157
BERT	10	2152	500	188.55	3.77	0.125
BERT + KGE	10	2152	500	188.50	3.77	0.126
ERNIE	10	2152	500	187.27	3.75	0.145
ERNIE + KGE	10	2152	500	187.27	3.79	0.146
T5	10	2152	500	189.48	3.83	0.154
T5 + KGE	10	2152	500	190.75	3.82	0.157

Topk: Top 5 or top 10 entity summarization
Input Triples: Number of triples that have to be scored as part of the benchmark dataset
Output Triples: Number of triples that ESLM outputs as part of the entity summarizations
Training Time Total: Total training time for all folds in seconds
Training Time Mean: Mean training time per epoch in seconds
Prediction Time Single Triples: Inference time for a single triple in seconds

Table 8: Analysis of computational efficiency experiment in Table 7

	Training Time		Prediction Time
	Total	Mean	Single Triples
Top 5	246.958	4.940	0.117
Top 10	244.231	4.884	0.120
BERT	240.520	4.809	0.107
ERNIE	234.018	4.682	0.121
T5	262.245	5.245	0.128
KGE	248.021	4.961	0.117
w/o KGE	243.168	4.863	0.120

Figure 2: In the following example, entity summarization is used to create a summary of Marie Curie's life as a scientist. In itself, the triple with the information that Marie Curie is married to Piere Curie seems insignificant for this summary. However, if one considers the other triples and thus, for example, the fact that the two conducted research together and achieved important successes, the significance of this triple for the entity summarization increases.

Context:
A summary of Marie Curie's research is to be created.

Triple to be scored:
(Marie Curie, Married To, Pierre Curie)

Other triples in entity description:
(Marie Curie, Collaborated With, Pierre Curie)
(Marie Curie, Discovered, Radium)
(Pierre Curie, Discovered, Radium)
...

Table 9: Mean and median of the node degree for DBpedia and for the entities for which ESLM performs worst for top 5 and top 10 summarization

Measure	**DBpedia**	**Worst Entities**	
	Overall	**Top 5**	**Top 10**
Node Degree Mean	35.49	44.73	51.92
Node Degree Median	31	39	50

Table 10: Mean and median of the node degree for the individual classes in DBpedia

Measure	**Agent**	**Work**	**Species**	**Location**	**Event**
Node Degree Mean	52.44	34.20	25.88	35.16	29.76
Node Degree Median	49	33	23	34	26

Figure 3: The figure shows the class distribution for the entities from DBpedia for which ESLM performs worst in the top 5 entity summarization. While in DBpedia overall the classes are equally distributed, you can see here that the agent class is overrepresented and the event class is underrepresented.

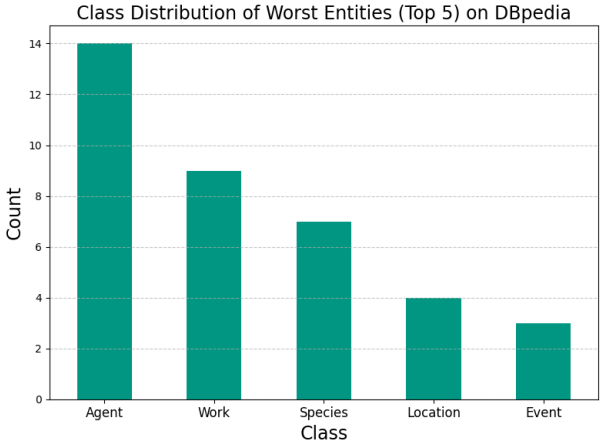

Table 11: NDCG scores for the ESLM baseline (T5 + KGE) and its modifications

Method	**DBpedia**		**LinkedMDB**		**FACES**		**Mean**
	$k = 5$	$k = 10$	$k = 5$	$k = 10$	$k = 5$	$k = 10$	
T5 + KGE	0.846	0.908	0.867	0.877	0.739	**0.795**	**0.839**
T5 + KGE + Modified Information Fusion	**0.852**	0.911	**0.868**	0.875	0.732	0.787	0.837
T5 + KGE + Second-level Encoder	0.830	0.888	0.801	0.867	0.672	0.741	0.800
T5 + KGE + Prompting	0.848	**0.913**	0.865	**0.879**	**0.740**	0.790	**0.839**

Figure 4: This example illustrates how the context string is created for any given entity. The last string consisting of context and triple string is passed to the language model encoder as input.

Entity:
Battle of Zacatecas (1914)

Class:
Event

Context String:
The entity Battle of Zacatecas (1914), a Event, is being summarized. How relevant is the following triple for this summary? [SEP]

Context String with Example Triple:
The entity Battle of Zacatecas (1914), a Event, is being summarized. How relevant is the following triple for this summary? [SEP] Battle of Zacatecas (1914) [SEP] isPartOfMilitaryConflict [SEP] Mexican Revolution

References

[1] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *NeurIPS* (2000).

[3] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. *NeurIPS* (2013).

[4] Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Firmansyah, A. F., Moussallem, D., and Ngomo, A.-C. N. Gates: using graph attention networks for entity summarization. In *Proceedings of the 11th Knowledge Capture Conference* (2021).

[6] Firmansyah, A. F., Moussallem, D., and Ngomo, A.-C. N. Eslm: Improving entity summarization by leveraging language models. In *The Semantic Web* (2024).

[7] Graves, A., and Schmidhuber, J. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks* (2005).

[8] Gunaratna, K., Thirunarayan, K., and Sheth, A. Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2015).

[9] Javadi, S., Moradan, A., Sorkhpar, M., Zaporojets, K., Mottin, D., and Assent, I. Wiki entity summarization benchmark. *arXiv preprint arXiv:2406.08435* (2024).

[10] Liu, Q., Cheng, G., Gunaratna, K., and Qu, Y. Esbm: an entity summarization benchmark. In *The Semantic Web - ESWC 2020* (2020).

[11] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR* (2020).

[12] Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223* (2019).

[13] Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence* (2020).

[14] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning* (2016).

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NeurIPS* (2017).

[16] Wei, D., Liu, Y., Zhu, F., Zang, L., Zhou, W., Han, J., and Hu, S. Esa: entity summarization with attention. *arXiv preprint arXiv:1905.10625* (2019).

[17] Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129* (2019).