

# ESA: Entity Summarization with Attention

Dongjun Wei\*  
Yaxin Liu\*  
weidongjun@iie.ac.cn  
liuyaxin@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China  
University of Chinese Academy of  
Sciences, Beijing, China

Fuqing Zhu†  
zhufuqing@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China

Liangjun Zang  
zangliangjung@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China

Wei Zhou  
zhouwei@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China

Jizhong Han  
hanjizhong@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China

Songlin Hu  
husonglin@iie.ac.cn  
Institute of Information  
Engineering, CAS, Beijing, China

## ABSTRACT

Entity summarization task aims at creating brief but informative descriptions of entities from Knowledge Graph. While previous work mostly focuses on traditional techniques such as clustering algorithms and graph models, we make an attempt to integrate deep learning methods into this task. In this paper, we propose an Entity Summarization with Attention (ESA) model, which is a neural network with supervised attention mechanisms for entity summarization. Specifically, we first calculate attention weights for facts in each entity. Then, we rank facts to generate reliable summaries. We explore techniques to solve complex learning problems presented by the ESA. On several benchmarks, experimental results show that ESA improves the quality of the entity summaries in both F-measure and MAP compared with some state-of-the-art methods, demonstrating the effectiveness of ESA. The source code and output can be accessed in <https://github.com/WeiDongjunGabriel/ESA><sup>1</sup>.

## CCS CONCEPTS

- **Computing methodologies** → *Semantic networks*.

## KEYWORDS

knowledge graph, entity summarization, neural network, supervised attention mechanism

## 1 INTRODUCTION

Since Knowledge Graph (KG) was first formally defined by Google in 2012, it has been widely applied to various communities in Artificial Intelligence (AI). KG serves for describing real-world entities and the relationship among entities. The way to represent databases in KG to describe entities is generally by Resource Description Framework (RDF), in

the form of  $\langle \text{subject}, \text{predicate}, \text{object} \rangle$  [6]. With knowledge databases rapidly growing up, the amount of entities and relations in KG simultaneously rises in an alarming rate. This phenomenon makes more challenge to extract or focus on considerable representative triples. To comprehend lengthy descriptions in large-scale KG quickly, summarizing useful information to condense the scale of knowledge databases is an emerging problem. Entity summarization is a method to extract both brief and informative entities, which has attracted keen interest in recent years. Since high quality of extracted entities is fundamental to derive subsequent knowledge in kinds of semantic tasks.

Cheng *et al.* [4] proposed RELIN to rank features based on relatedness and informativeness for quick identification of entities, which is adapted according to random surfer model. DIVERSUM [13] takes the diverse property of entities into consideration for summarizing tasks in KG. FACES [8] makes a proper balance between centrality and diversity of extracted triples through Cobweb algorithm. FACES-E proposed by Gunaratna *et al.* [9] optimizes FACES by considering the effect of literals in entity summarization. CD [17] follows the idea of binary quadratic knapsack problem to complete entity summarization. Based on PageRank algorithm to rank triples, LinkSUM [14] focuses more on the objects rather than utilizing the diversity of properties.

In retrospect, previous work requires considerable prior knowledge to construct complex ranking rules for entity summarization. Besides, we can hardly find deep learning methods for entity summarization. Due to attention mechanism generates different weights according to human concern, we can acquire higher weights for triples that people more focus on. Following the advantages of BiLSTM, contextual information is fully used to capture more informative triples. Therefore, we propose a model called ESA, which uses supervised attention mechanism with BiLSTM. The ESA allows us to calculate attention weights for triples derived from each entity, final reliable summaries can be extracted by ranking weights.

\*Both authors contributed equally to this research.

†Corresponding Author: Fuqing Zhu. E-mail: zhufuqing@iie.ac.cn

<sup>1</sup>Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

## 2 TASK DESCRIPTION

RDF is an abstract data model, and an RDF graph consists of a collection of statements. Simple statements generally represent real-world entities, which are usually stored as triples. Each triple  $t$  represents a fact that is in the form of  $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ , denoted as  $\langle s, p, o \rangle$ . Since RDF data is encoded by unique identifiers (URIs), an entity in RDF graphs can be regarded as a subject with all predicates and corresponding objects to those predicates.

**Definition 1 (Entity Summarization):** Entity Summarization (ES) is a technique to summarize RDF data for creating concise summaries in KG. The subject of each entity provides the core for summarizing entities. Therefore, the task of entity summarization is defined as extracting a subset from a lengthy feature set of each entity with the respective subject. Given an entity  $e$  and a positive integer  $k$ , the output is  $\text{top-}k$  features of every entity  $e$  in the ranking list of  $ES(e, k)$ .

## 3 THE PROPOSED ESA

We model ES as a ranking task similar to existing work, such as RELIN, FACES, and ES-LDA. Different from the traditional approaches of generating entity summaries in KG, the ESA is a neural network model using sequence model, Figure 1 describes the architecture of the model.

Similar to most sequence models [5], the ESA has an encoder-decoder structure. The encoder is consisted of knowledge representation and BiLSTM, which maps an input sequence  $(t_1, t_2, \dots, t_n)$  of RDF triples from a certain entity to a continuous representation  $\mathbf{h} = (h_1, h_2, \dots, h_n)$ . The decoder is mainly composed of attention model. Given  $\mathbf{h}$ , the decoder then uses a supervised attention mechanism generates an output vector  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$  representing attention vector for each entity, which is then used as evidence for summarizing entities. Higher attention weights are related to more important triples, we finally select triples according to  $\text{top-}k$  highest weights as our entity summaries.

### 3.1 Knowledge Representation

Entities in large-scale KG are usually described as RDF triples, while each triple consists of a subject, a predicate, and an object. MPSUM proposed by Wei *et al.* [16] takes the uniqueness of predicates and the importance of objects into consideration for entity summarization. The experimental results show that the characteristics of predicates and objects are key factors to select entities. In order to make full use of the information contained by RDF triples, we extract predicates and objects from above triples. Let  $n$  be the number of triples with the same subject  $s$ , then two lists respectively based on extracted predicates and objects are  $l_1 = (p_1, p_2, \dots, p_n)$  and  $l_2 = (o_1, o_2, \dots, o_n)$ , where  $p_i$  and  $o_i$  are corresponding predicates and objects from the  $i$ -th triples. For each entity, we employ different methods to map predicates and objects into continuous vector space respectively [10]. To solve the UNK problem of objects, we

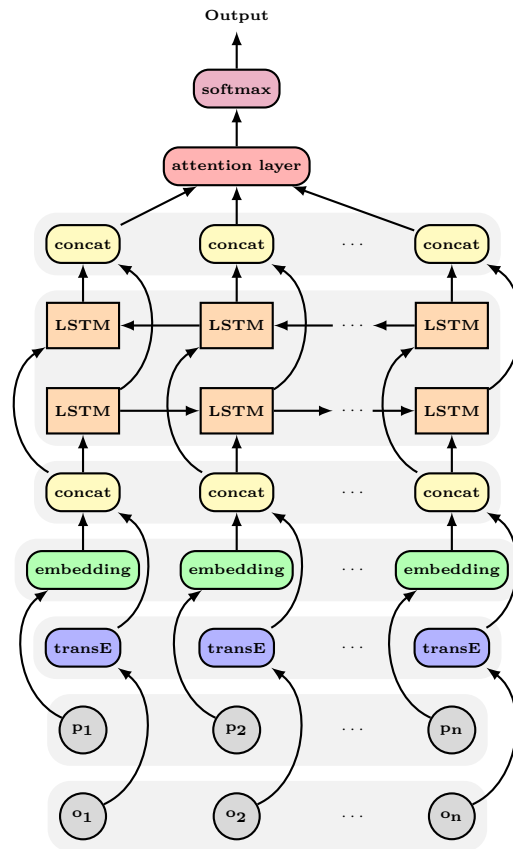


Figure 1: The Architecture of ESA Model

employ different methods for each entity to map predicates and objects into continuous vector space respectively.

**Predicate Embedding Table.** We use learned embeddings [1] to convert a predicate input to a vector of dimension  $d_p$ . We randomly initialize embedding vector for each predicate and tune it in training phase.

**Object Embedding Table.** Unlike generating representation of predicates based on word embedding technique, we use TransE model [2] to map objects into vectors of dimension  $d_o$ . We first pretrain TransE model based on ESBM benchmark v1.1, and extract the word vectors of objects to construct a lookup table for object vectors. Then we obtain object vectors by looking up the table as input, the object vectors are fixed during training.

### 3.2 BiLSTM Network

We first randomly map the set of triples into a sequence, then we employ BiLSTM to extract the information of former triples from 1 to  $i - 1$  and later triples from  $i + 1$  to  $n$ , where the information respectively propagations forward and backward. In this paper, we denote the  $LSTM_L$  and  $LSTM_R$  as the forward and backward LSTM model,  $x_i$  as the input at the time step  $i$  for  $LSTM_L$  and  $LSTM_R$ , the

corresponding of output of them are  $\mathbf{h}_{L_i}$  and  $\mathbf{h}_{R_i}$ . We encode the input  $x_i$  using Bidirectional LSTM as follows:

$$\begin{aligned} \mathbf{h}_{L_i} &= LSTM_L(x_i, \mathbf{h}_{L_{i-1}}) \\ \mathbf{h}_{R_i} &= LSTM_R(x_i, \mathbf{h}_{R_{i-1}}). \end{aligned} \quad (1)$$

The final output  $\mathbf{h} = (h_1, h_2, \dots, h_n)$ , and its component  $\mathbf{h} = (h_1, h_2, \dots, h_n)$  of BiLSTM is calculated by concatenating  $\mathbf{h}_{L_i}$  and  $\mathbf{h}_{R_i}$ .

Moreover, the  $\mathbf{h}_s$  is concatenated by  $\mathbf{h}_s^1$  and  $\mathbf{h}_s^2$ .  $\mathbf{h}_s^1$  is the value of hidden state from the final cell of upper LSTM layer, while  $\mathbf{h}_s^2$  is the value of hidden state from the final cell of lower LSTM layer. We then take  $\mathbf{h}_s$  as the input of subsequent attention layer.

### 3.3 Supervised Attention

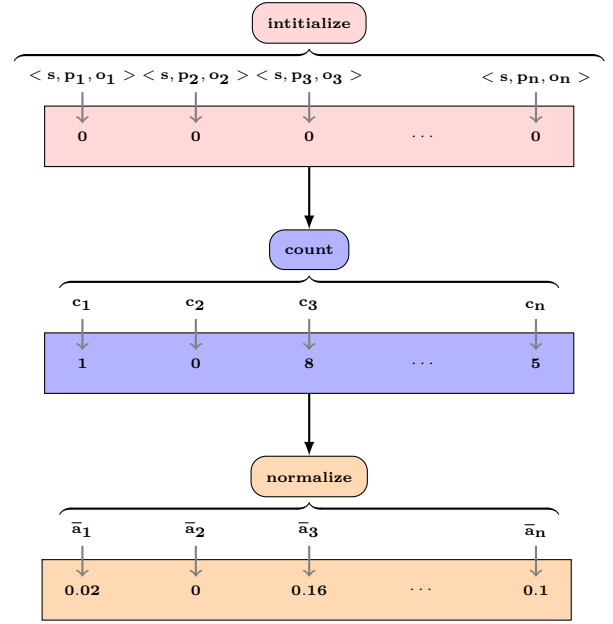
Attention model is a mainstream neural network in various tasks such as Natural Language Processing (NLP) [18] [15]. For instance, in machine translation tasks [11], only certain words in the input sequence may be relevant for predicting the next [3]. Attention model incorporates this notion by allowing the model to dynamically pay attention to only certain parts of the input that help in performing the task at hand effectively. In entity summarization task, when users observe the facts in each subject, they may pay more attention to certain facts than the rest, which can be modeled based on Attention model by assigning an attention weight for each fact in the subject.

Given above considerations, we first construct gold attention vectors using existing datasets. Then, we employ attention mechanism to generate machine attention vectors.

**Gold Attention Vectors.** In this work, we use ESBM benchmark v1.1 as our dataset. For each subject, we need to summarize, ESBM benchmark v1.1 not only provides the whole RDF triples which are related to this subject, but also provides several sets of *top-5* and *top-10* triples selected by different users according to their preference. Above triples can be utilized to construct gold attention vectors, which we can utilize to construct gold attention vectors. We first initialize an attention vector to zero of dimension  $n$ , where  $n$  is the number of RDF triples corresponding to a specific subject. Then, we count the frequency of each triple selected by users to update the vector, the  $i$ -th value  $c_i$  in this vector represents the frequency of triple  $t_i$ . Since ESBM benchmark v1.1, each subject has five sets of *top-5* and *top-10* triples selected by five different users, so the frequency of each triple ranges from 0 to 5. Figure 2 illustrates the details, where  $\bar{\alpha}$  is the final gold attention vector after normalization, each value in  $\bar{\alpha}$  is calculated by the following equation,  $\bar{\alpha}_i$  denotes the  $i$ -th value in vector  $\bar{\alpha}$ :

$$\bar{\alpha}_i = \frac{c_i}{\sum_{i=1}^n c_i}. \quad (2)$$

**Machine Attention Vectors.** To generate machine attention vectors with Attention model, we first obtain the output vectors  $\mathbf{h} = (h_1, h_2, \dots, h_n)$  that the BiLSTM layer produced. Then, the attention layer can automatically learn attention vector



**Figure 2: The Process of Constructing of Gold Attention Vectors**

$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$  based on  $\mathbf{h}$ . We use softmax function to generate final attention vector  $\alpha$ :

$$\alpha = \text{softmax}(\mathbf{h}_s^T \mathbf{h}), \quad (3)$$

where  $\mathbf{h}_s$  is concatenated by  $\mathbf{h}_s^1$  and  $\mathbf{h}_s^2$ ,  $\mathbf{h}_s^1$  is the value of hidden state from the final cell of upper LSTM layer, while  $\mathbf{h}_s^2$  is the value of hidden state from the final cell of lower LSTM layer. We rank final attention weight vector  $\alpha$ . Then we obtain the entity summaries based on corresponding *top-k* values.

**Training.** Given the gold attention  $\bar{\alpha}$  and the machine attention  $\alpha$  produced by our model, we employ cross-entropy loss and define the loss function  $L$  of the proposed ESA model as follows:

$$L(\alpha, \bar{\alpha}) = \text{CrossEntropy}(\alpha, \bar{\alpha}). \quad (4)$$

Finally, we use back-propagation algorithm to jointly train the ESA model.

## 4 EXPERIMENT

In this section, we first introduce the datasets and evaluation metrics employed in our experiment. Then we give the implementation details to describe the overall procedure. To prove the effectiveness of our model, we finally compare ESA with the state-of-the-art approaches, including RELIN [4], DIVERSUM [13], CD [17], FACES [8] FACES-E [8], and LinkSUM [14]. The experimental results are presented in section 4.2.

In this work, experiments are conducted based on ESBM Benchmark v1.1 as ground truth. The ESBM benchmark

v1.1 consists of 175 entities including 125 entities are from DBpedia<sup>2</sup> and the rest entities are from LinkedMDB<sup>3</sup>. The datasets and ground truth of the entity summarization can be obtained from the ESBM<sup>4</sup>.

#### 4.1 Implementation Details

We apply word embedding technique to map predicates into continuous space and use pretrained translation vectors with TransE for objects. We first randomly partition the ESBM dataset into five subsets for cross-validation before training. During training, the word vectors of predicates are jointly trained while the object vectors are fixed. The whole ESBM benchmark v1.1 is trained using thunlp<sup>5</sup>. We generate gold attention vectors based on ESBM benchmark v1.1, and calculate machine attention vectors based on our model. Finally, we compare our model in terms of *top-5* and *top-10* entity summaries with the benchmark results of the entity summarization tools, *i.e.*, RELIN, DIVERSUM, CD, FACES-E, FACES, and LinkSUM, as shown in Table 1 and Table 2.

Hyper-parameters are tuned on the selected datasets. We set the dimension of predicate embedding to 100, the dimension of TransE to 100. The initial learning rate in our model is set to 0.0001, which is an invariant parameter during training.

#### 4.2 Experimental Results

In this paper, we have carried out several experiments using F-measure and MAP metrics based on two datasets: DBpedia and LinkedMDB. The results regarding F-measures are shown in Table 1, and MAP are shown in Table 2. The results regarding F-measure and MAP are respectively shown in Table 1 and Table 2. ESA achieves better results than all other state-of-the-art approaches in each dataset, as well as performs best in each metric.

**F-measure.** As shown in Table 1, the best improvement in single dataset is under *top-5* summaries generated from DBpedia, our model reaches the highest F-measure with 0.310, which exceeds the previously best result produced by CD. In terms of DBpedia dataset, the total increase of *top-5* and *top-10* summaries is 3.1%. For LinkedMDB dataset, our model obtains the best score in both  $k = 5$  and  $k = 10$ . Meanwhile, we combine two datasets to implement entity summarization, our model has 7.96% and 5.82% increase respectively for the results based on *top-5* and *top-10* results.

**MAP.** Our model also achieves better scores for MAP metric, as Table 2 shows, where the best increase is 3% represented in LinkedMDB for  $k = 10$ . The improvement of LinkedMDB is more obvious in MAP metric than F-measure, where the total increase is up to 5.6%.

**ALL.** Combining Table 1 and Table 2, it is evident that our ESA yields better results both for F-measures and MAPs. It

is worth mentioning that our model outperforms all other state-of-art approaches in both F-measure and MAP given by ESBM benchmark v1.1, which can significantly demonstrate the effectiveness of our model.

## 5 CONCLUSION

In this work, we propose ESA, a neural network with supervised attention mechanisms for entity summarization. Our model aims at involving the human preference to augment the reliability of extracted entities. Meanwhile, we explore the way to construct gold attention vectors for modelling supervised attention mechanism. The ESA applies extracted predicates and objects as input, in particular, we exploit different but proper knowledge embedding methods respectively for predicates and objects, where the word embedding method is for predicates and TransE is for objects. The final output of ESA is normalized attention weights, which can be used to select representative entities. Our experiments indicate that word embedding technique and graph embedding technique like TransE can be combined together into a single task, which can better represent the fact or knowledge in knowledge graph and provide a more powerful input vectors for neural networks or other models. Experimental results show that our work outperforms all other approaches both in F-measure and MAP.

## 6 FUTURE WORK

In future work, we expect to integrate various deep learning methods, and design several more powerful and effective neural networks. Specifically, we may improve our work in the following ways: (1) extending the scale of training set to better train our models; (2) instead of employing TransE model to tackle the UNK problem, we plan to analyze RDF triples in more fine-grained aspects.

## REFERENCES

- [1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2000. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2000), 1137–1155.
- [2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*.
- [3] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. 2019. An Attentive Survey of Attention Models. *CoRR* abs/1904.02874 (2019).
- [4] Gong Cheng, Thanh Tran, and Yuzhong Qu. 2011. RELIN: Relatedness and Informativeness-Based Centrality for Entity Summarization. In *Semantic Web-iswc -international Semantic Web Conference*.
- [5] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *SSST@EMNLP*.
- [6] Maria De-Arteaga, Alexey Romanov, Hanna M. Wallach, Jennifer T. Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Cem Geyik, Krishnamurthy Kenthapadi, and Adam Tauman Kalai. 2019. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In *FAT*.
- [7] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Amit Sheth, and Cheng Gong. 2016. *Gleaning Types for Literals in RDF Triples with Application to Entity Summarization*.
- [8] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P. Sheth. 2015. FACES: Diversity-Aware Entity Summarization Using Incremental Hierarchical Conceptual Clustering. In *AAAI*.

<sup>2</sup><https://wiki.dbpedia.org>

<sup>3</sup><http://linkedmdb.org>

<sup>4</sup><http://ws.nju.edu.cn/summarization/esbm/>

<sup>5</sup><https://github.com/thunlp/TensorFlow-TransX>

|                      | DBpedia      |              | LinkedMDB    |              | ALL          |              |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                      | k=5          | k=10         | k=5          | k=10         | k=5          | k=10         |
| <b>RELIN</b> [4]     | 0.242        | 0.455        | 0.203        | 0.258        | 0.231        | 0.399        |
| <b>DIVERSUM</b> [13] | 0.249        | 0.507        | 0.207        | 0.358        | 0.237        | 0.464        |
| <b>CD</b> [12]       | 0.287        | 0.517        | 0.211        | 0.328        | 0.252        | 0.455        |
| <b>FACES–E</b> [7]   | 0.280        | 0.485        | 0.313        | 0.393        | 0.289        | 0.461        |
| <b>FACES</b> [8]     | 0.270        | 0.428        | 0.169        | 0.263        | 0.241        | 0.381        |
| <b>LinkSUM</b> [14]  | 0.274        | 0.479        | 0.140        | 0.279        | 0.236        | 0.421        |
| <b>ESA</b>           | <b>0.310</b> | <b>0.525</b> | <b>0.320</b> | <b>0.403</b> | <b>0.312</b> | <b>0.491</b> |

Table 1: Comparison of F-measure on ESBM benchmark v1.1

|                      | DBpedia      |              | LinkedMDB    |              | ALL          |              |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                      | k=5          | k=10         | k=5          | k=10         | k=5          | k=10         |
| <b>RELIN</b> [4]     | 0.342        | 0.519        | 0.241        | 0.355        | 0.313        | 0.466        |
| <b>DIVERSUM</b> [13] | 0.310        | 0.499        | 0.266        | 0.390        | 0.298        | 0.468        |
| <b>CD</b> [12]       | -            | -            | -            | -            | -            | -            |
| <b>FACES–E</b> [7]   | 0.388        | 0.564        | 0.341        | 0.435        | 0.375        | 0.527        |
| <b>FACES</b> [8]     | 0.255        | 0.382        | 0.155        | 0.273        | 0.227        | 0.351        |
| <b>LinkSUM</b> [14]  | 0.242        | 0.271        | 0.141        | 0.279        | 0.213        | 0.345        |
| <b>ESA</b>           | <b>0.392</b> | <b>0.582</b> | <b>0.369</b> | <b>0.465</b> | <b>0.386</b> | <b>0.549</b> |

Table 2: Comparison of MAP on ESBM benchmark v1.1

- [9] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Amit P. Sheth, and Gong Cheng. 2016. Gleaning Types for Literals in RDF Triples with Application to Entity Summarization. In *ESWC*.
- [10] Siwei Lai, Kang Liu, Liheng Xu, and Jian Zhao. 2016. How to Generate a Good Word Embedding. *IEEE Intelligent Systems* 31 (2016), 5–14.
- [11] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.
- [12] Danyun Xu Liang Zheng Yuzhong Qu. 2016. *CD at ENSEC 2016: Generating Characteristic and Diverse Entity Summaries*.
- [13] Marcin Sydow, Mariusz Pikula, and Ralf Schenkel. 2010. DIVERSUM: Towards diversified summarisation of entities in knowledge graphs. *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)* (2010), 221–226.
- [14] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. 2016. LinkSUM: Using Link Analysis to Summarize Entity Data. In *ICWE*.
- [15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. (2017).
- [16] Dongjun Wei, Shiyuan Gao, Yaxin Liu, Zhibing Liu, and Longtao Huang. 2018. MPSUM: Entity Summarization with Predicate-based Matching. In *EYRE*.
- [17] Danyun Xu, Liang Zheng, and Yuzhong Qu. 2016. CD at ENSEC 2016: Generating Characteristic and Diverse Entity Summaries. In *SumPre@ESWC*.
- [18] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine* 13 (2018), 55–75.