Wikipedia Infobox Type Prediction Using Embeddings

Russa Biswas[1,2], Rima Türker[1,2], Farshad Bakhshandegan-Moghaddam[1,2], Maria Koutraki[1,2], and Harald Sack[1,2]

[1] FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
[2] Karlsruhe Institute of Technology, Institute AIFB, Germany
{firstname.lastname}@fiz-karlsruhe.de
{firstname.lastname}@kit.edu

Abstract. Wikipedia, the multilingual, free content encyclopedia has evolved as the largest and the most popular general reference work on the Internet. Since the time of commencement of Wikipedia, crowd sourcing of articles has been one of the most salient features of this open encyclopedia. It is obvious that enormous amount of work and expertise goes in the creation of a self-content article. However, it has been observed that the infobox type information in Wikipedia articles is often incomplete, incorrect and missing. This is due to the human intervention in creating Wikipedia articles. Moreover, the type of the infoboxes in Wikipedia plays a vital role in the determination of RDF type inference in the Knowledge Graphs such as DBpedia. Hence, there arouses a necessity to have the correct infobox type information in the Wikipedia articles. In this paper, we propose an approach of predicting Wikipedia infobox type information using both word and network embeddings. Furthermore, the impact of using minimalistic information such as *Table of Contents* and *Named Entity* mentions in the abstract of a Wikipedia article in the prediction process has been analyzed as well.

Keywords: Wikipedia, Infobox, Embeddings, Knowledge Graph, Classification

1 Introduction

Since the commencement of Wikipedia, it has emerged as the largest multilingual encyclopedias available on the Internet. It is the most widely used general reference non-profit crowd sourcing project, owned by the Wikimedia Foundation[3]. A huge amount of expertise and effort is involved in the creation of Wikipedia articles. Wikipedia articles are generated as an amalgamation of the information contributed by humans in all the different segments of the article layout. A typical Wikipedia article comprises both structured and unstructured data. The unstructured data consists of the text describing the article content whereas, structured data is represented in the form of an *infobox* containing property

[3] https://en.wikipedia.org/wiki/Wikipedia

value pairs summarizing the content of the article. An infobox is a fixed-format table usually added to consistently present a summary of some unifying aspects that the articles share and sometimes to improve navigation to other interrelated articles[4]. Furthermore, the structured data present in the infoboxes of the Wikipedia articles are widely used in different Knowledge Graphs (KGs) such as DBpedia, Google's Knowledge Graph, Microsoft Bing's Satori etc. [5].

The selection of the infobox type is determined collaboratively through discussion and consensus among the editors. The infobox types or templates are created and assigned based on categorical type of the articles, i.e. the same template should be assigned to similar articles. However, no integrity tests are conducted to check the correctness of the infobox assignments, leading to erroneous infobox types [10,11,12]. For instance, *George H. W. Bush* and *George W. Bush* both are former Presidents of the USA, but they have different infobox types assigned to their respective Wikipedia articles. The former has the infobox type of *office holder* whereas the later has the infobox type *president*. Additionally, it is not mandatory to select an infobox type for the creation of an article. Thus, about 70% of the Wikipedia articles do not contain an infobox. It has been observed that infoboxes are missing for newer articles or articles on less popular topics.

Moreover, RDF type information in KGs such as DBpedia is derived directly from Wikipedia infobox types by automated information extraction. Therefore, the completeness and correctness of the infobox type information is of great importance. Different studies [10,11,12] have strengthened the fact that the infobox type information is often noisy, incomplete and incorrect. However, the infobox type prediction problem in Wikipedia can be viewed as a text classification problem with infobox types as labels.

In this paper, we present a novel approach to predict Wikipedia infobox types by using word embeddings on the text present in the *Table of Contents(TOC)*, the article's *abstract*, and additionally network embeddings on the *Named Entities* mentioned in the abstract of the article. The TOC consists of the headings and the subheadings of the article text, summarizing the information content of the text in the section underneath. To the best of our knowledge so far, Wikipedia infobox types have not been predicted using different types of embeddings including TOC as one of the features. In this paper, the impact of using a minimalistic yet informative feature such as the TOC in the classification process via classical and neural network based classifiers is studied. Additionally, the importance of Named Entities mentioned in the abstract of the article to determine the infobox types has been analyzed.

The rest of the paper is structured as follows. To begin with, a review of the related work is provided in Section 2 followed by a short description of the approach in Section 3. Section 4 accommodates the outline of the experimental setup followed by a report on the results in Section 5. Finally, an outlook of future work is provided in Section 6.

[4] https://en.wikipedia.org/wiki/Help:Infobox

2 Related Work

Scope. The aim of this work is to predict the infobox types by leveraging the *Table of Contents*, *abstract* and the *Named Entity* mentions in the abstract of the Wikipedia articles. This section presents prior related work on *infobox type prediction*. RDF type information in DBpedia is derived from Wikipedia infobox type information. Therefore, the Wikipedia infobox type prediction problem can be seen as a closely related task of RDF type prediction, which is covered first in the subsequent section followed by Wikipedia infobox type prediction.

RDF Type Prediction. A statistical heuristic link based type prediction mechanism, SDTyped, has been proposed by Paulheim et al. and was evaluated on DBpedia and OpenCyc [8]. Another RDF type prediction of KGs has been studied by Melo et al., where the type prediction of the KGs is performed via the hierarchical SLCN algorithm using a set of incoming and outgoing relations as features for classification [6]. Kleigr et al.[4] proposed a supervised hierarchical SVM classification approach for predicting the RDF types in DBpedia by exploiting the contents of Wikipedia articles.

As already mentioned, these approaches infer DBpedia RDF type information of entities by taking into account properties present in DBpedia or Wikipedia content. On the contrary, we intend to predict the Wikipedia infobox types by considering the TOC, abstract and Named Entities. Hence, these are worth mentioning but different from the proposed work of this paper.

Wikipedia Infobox Type Prediction. One of the initial works in this domain was proposed by Wu et al.[11]. They presented KYLIN, a prototype which automatically creates new infoboxes and updates the existing incomplete ones. To do so, KYLIN takes into account pages having similar infoboxes, determines the common attributes in them to create training examples, followed by learning a CRF extractor. KYLIN also automatically identifies missing links for proper nouns on each page, resolving each to a unique identifier.

Sultana et al.[10] focuses on automated Wikipedia infobox type prediction by training a SVM classifier on the feature set of TF-IDF on the first k sentences of an article as well as on categories and Named Entity mentions.

Yus et al.[12] introduced a tool based on Semantic Web technologies which uses statistical information of Linked Open Data(LOD) to create, update and suggest infoboxes to users during creation of a Wikipedia article. Bhuiyan et al.[1] presented an automated NLP based unsupervised infobox type prediction approach by exploiting the hyponyms and holonyms in Wikipedia articles.

In contrast to the aforementioned works, we propose a classification based infobox type prediction approach by combining word embeddings and network embeddings to generate feature vectors instead of TF-IDF. The proposed method does not focus on creating new infobox templates rather it focuses on correcting and complementing the missing infoboxes in the articles. Furthermore, unlike prior works, the TOC is also considered as one of the features for the classification process.

3 Infobox Type Prediction

This section contains a detailed explanation of the workflow and the methodologies used for the multi-label classification process to predict the infobox type information. The workflow is illustrated in Figure 1.

3.1 Features

Three different features are extracted from the Wikipedia articles for the classification task:

- **TOC**[5] which is automatically generated based on the section and subsection headers of the Wikipedia articles depicting a summarization of the content in a single word or a short sentence.
- **Abstract *(A)*** of the Wikipedia articles i.e. the summary of the entire article content.
- **Named Entities *(E)*** present in the abstract section of the articles are most likely to be related to the article hence assumed to provide more information in the classification process. The internal hyperlinks within Wikipedia are used to identify the Named Entities mentioned in the abstract.

3.2 Embeddings

Both word and network embeddings are used to generate features for the classifiers.

Word2Vec. Word2vec [7] aims to learn the distributed representation for words reducing the high dimensional word representations as well as categorize semantic similarities between them in large samples of text. The semantic similarities of the linguistic terms based on their distribution in the TOCs as well as in the abstract of different types of Wikipedia articles is vital. In this paper, the Google pre-trained word vectors[6] are used to generate word vectors for each word present in the TOC and the abstract. Google pre-trained word2vec model includes word vectors for a vocabulary of three million words and phrases trained on roughly 100 billion words from a Google News dataset. The vector length has been restricted to 300 features. It is to be noted that the information in TOC has been considered as free text in this work.

RDF2Vec. RDF2Vec [9] is an approach of latent representations of entities of a KG into a lower dimensional feature space with the property that semantically similar entities appear closer to each other in the feature space. In this work, the pre-trained RDF2Vec uniform model vectors from DBpedia[7] have been used

[5] https://en.wikipedia.org/wiki/Help:Section
[6] https://code.google.com/archive/p/word2vec/
[7] http://data.dws.informatik.uni-mannheim.de/rdf2vec/models/DBpedia/2016-04/

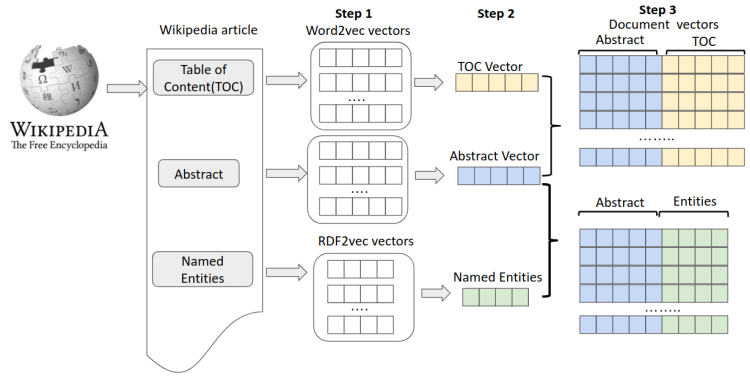

Fig. 1. Workflow of the feature extraction from the Wikipedia articles (best viewed in color)

to extract the vectors for the Named Entities mentioned in the abstract of the Wikipedia article. The vector length has been restricted to 200 features. Similar to the word2vec word vectors, these vectors are generated by learning a distributed representation of the entities and their properties in the underlying KG. The intuition behind incorporating the vectors of Named Entities mentioned in the abstract of Wikipedia articles into the feature set is to include the features or the properties of the different entities from the DBpedia KG into the classification process.

3.3 Feature Vectors

The feature vectors for each of the Wikipedia articles are generated using the following steps:

Step 1: Extract *word vectors* for each word in the TOC as well as the abstract from the Google pre-trained model. Also, extract *entity vectors* for the Named Entities mentioned in the abstract from the RDF2Vec pre-trained model of DBpedia version 2016-04.

Step 2: Generate an *abstract vector* for each document by performing vector addition on all the word vectors of the abstract and normalize by the total number of words present in the abstract. Similarly, *TOC vectors* and *entity vectors* are also generated.

Step 3: Generate *document vectors* - Two sets of document vectors are generated for the training of two classifiers. The abstract vector of each document is concatenated separately with the TOC vector and entity vector of the corresponding document to generate the document vectors.

3.4 Classification

As already discussed, the Wikipedia infobox type prediction problem can be reduced to a classification of the Wikipedia articles with word vectors and entity

vectors as features. In this work, we have trained the Wikipedia articles using two classifiers: *Random Forest (RF)* and *Multilabel Convolutional Neural Network (CNN)*. For the Random Forest classifier, the aforementioned document vectors coupled with the labels of the infobox types are used to train the classifier. Random Forest, as an ensemble method is less likely to overfit. Moreover, the subsets of the training set for bagging reduces the effects of the outliers in the data, if any. On the other hand, for CNN, the concept of sentence level classification task as discussed in [3] has been used for the classification process. The Google pre-trained word vectors are used to generate the vectors in the embedding layer followed by a fully connected softmax layer, whose output is the probability distribution over the labels. A detailed description of the experimental setup is provided in the following section.

4 Experimental Setup

This section contains a detailed explanation of the dataset followed by the generation of ground truth. For the Random Forest classifier, python scikit-learn[8] library has been used. For the CNN model, to classify the Wikipedia articles, based on the sentence classification concept as described in [3], TensorFlow version 1.0 has been used to build the model[9].

4.1 DataSet

Wikipedia 2016[10] version and RDF2Vec pre-trained model for DBpedia version 2016-04 have been used for this work. This version contains around 2000 different Wikipedia Infobox types. It has been observed that the frequency distribution of the Wikipedia Infobox types follows Zipf's law as shown in Figure 2. The x-axis represents the infobox types in numbers and y-axis represents the count of entities per infobox type.

More than half of all Wikipedia pages do not contain infobox types. Also, there are articles in Wikipedia containing more than one infobox type. However, in this work we considered only those articles having a single infobox type. The statistics of the Wikipedia articles with infobox types is provided in Table 1. It consists of the count of the Wikipedia articles containing TOC, abstract, infoboxes and the combination of these three together. Wikipedia redirect pages, disambiguation pages and the list pages are ignored in the dataset.

In this work, based on the popularity of the infoboxes, we have considered top 30 infobox types as labels with 5000 articles per label to train the classifiers. These 5000 articles per infobox type are selected by random sampling without replacement and the experiments are being carried out with three such datasets of the same size. It is important to note that the experiments are carried out to study the impact of the TOC and Named Entities separately as well as in combination with the abstract.

[8] http://scikit-learn.org/stable/
[9] https://github.com/yuhaozhang/sentence-convnet
[10] http://downloads.dbpedia.org/2016-10/core-i18n/en/pages_articles_en.xml.bz2

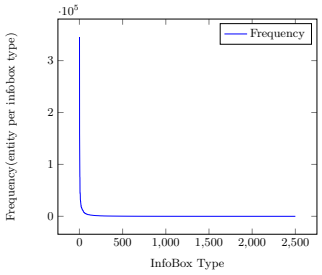

Fig. 2. Distribution of the number of entities per infobox type

Features	#Wikipedia articles
TOC	9,959,830
Abstract(A)	4,935,279
Infobox(I)	2,626,841
TOC + A + I	2,575,966

Table 1. Statistics of Wikipedia Articles

4.2 Train/Test

For the Random Forest classifier, the experiments are carried out for both 5-Fold Cross Validation (CV) as well as split with 80% data as train set and 20% data as test set. Identically, for the CNN 80% data is considered as train set and 20% data as test set. The results are discussed in the next section.

4.3 Ground Truth

Cross-Validation techniques are adequate, testing the proposed approach over unseen data reveals the generalization of the model and its robustness. To the best of our knowledge no benchmark exists in the field of Wikipedia infobox type prediction. Therefore, automated Ground Truth has been generated for the purpose. The manual creation of ground truth has the advantage of yielding benchmarks incorporating human knowledge on the topic. On the other hand, it incorporates significant disadvantages in terms of huge manual effort leading to a very small amount of ground truth generated. Therefore, in this work, the focus was to automatically generate ground truth with preserving the characteristics of the manually credited one. To do so, first, all the articles without infoboxes were extracted from Wikipedia version 2016[11]. Second, these articles were checked against the latest Wikipedia version 2018[12] to find if there existed an infobox type in the new version for them. This approach leads to the generation of ground truth comprising of 32000 Wikipedia articles in total.

4.4 Baseline

TF-IDF is one of the most widely used method to generate the vectors for the text classification problem [2]. As the infobox type prediction problem can be reduced to a classification problem, in this study, TF-IDF is considered as a baseline.

[11] http://downloads.dbpedia.org/2016-10/core-i18n/en/pages_articles_en.xml.bz2
[12] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

5 Results and Discussion

The experiments establish the fact that the text present in the TOC of a Wikipedia article is very minimal yet informative and its contribution in the classification process is not insignificant. With the CNN classifier, it reaches an accuracy of 76.5% micro-F1 score and around 65% with Random Forest. On the other hand, with the TF-IDF vectorizer the accuracy is very little. This is due to the fact that TF-IDF captures how important is the word for a document whereas the word embeddings capture the semantic similarity of the words in the text. Since the TOC is generated from the headings provided by the authors at the time of the creation of the articles and no guidelines being available, the vocabulary of the headings varies from article to article. Hence TF-IDF fails to capture the semantic similarity between them. For instance, in the Wikipedia article for *Bill Gates*, his early life is described under the section named *'Early Life'* whereas for *Steve Jobs*, his early life is described under the section *'Background'* and subsections *'Biological and adoptive family'* and *'Birth'*. TF-IDF fails because it treats them as different words. On the other hand, *'family'* appears in the top-5 similar words for *'life'* using the Google pre-trained vectors. Hence, the semantic similarity of the words is considered. This also explains the reason behind using the pre-trained vectors from Google. Therefore, it can be inferred that using word embeddings over minimalistic yet informative key words from an article is capable of predicting Wikipedia infoboxes.

However, using the abstracts as a feature to determine infobox types improves the quality of results both for TF-IDF as well as for word embeddings. A huge improvement is noticed over the TF-IDF approach due to the fact that long texts improve the quality of the vectors by this process. Furthermore, embeddings still work better for the classification process capturing the semantic similarities in text.

Combining both the features, abstract and TOC together provides the best result for all the cases. The data is less sparse now for TF-IDF to work better as compared to only considering TOC. Also, there is an improvement of 2% in the micro F1 score for the random forest and 1% increase in the CNN classifier from using only abstract. It can be inferred for the articles with very less text in the abstract but a well sectioned article for the rest of the document, TOC can be considered to play a vital role in the prediction process.

Additionally, for all the combination of the feature set as explained in Table 2, CNN performed better than Random Forest proving the fact that CNN models are trained better for large datasets. We also carried our experiments with SVM for a couple of the aforementioned feature sets and obtained similar results as with the Random Forest classifier. However, multi-class SVM classification with one vs. all approach was computationally expensive as compared to random forest for this task.

The impact of Named Entities in the classification process has also been studied as shown in Table 3. It is to be noted that with only the Named Entities in the abstract, the classification process is not as effective achieving a micro F1 score of around 45% with Random Forest Classifier and 62% with CNN model.

Feature Set	With Embedding			TF-IDF	
	RF(CV)	RF(Split)	CNN	RF(CV)	RF(Split)
Table of Contents	65%	65.8%	**76.5%**	38%	32.3%
Abstract	86%	86.4%	**95.1%**	80%	80.4%
Named Entities	45%	45.6%	**62%**	-	-
Table of Contents + Abstract	88%	88%	**96.1%**	83%	83.9%

Table 2. Performance of classifiers using micro F1 score over the feature sets related to Table of Contents

Feature Set	With Embedding	
	RF(CV)	RF(Split)
Named Entities	45%	45.6%
Abstract	86%	86.4%
Named Entities + Abstract	86%	86%
Named Entities + Abstract + Table of Contents	**87%**	**87.6%**

Table 3. Performance of classifiers using micro F1 score over the feature sets related to Named Entities

However, the addition of abstract word vectors to the entity vectors of an article leads to an improvement of up to 86%. Moreover, the combination of the abstract, TOC and the Named Entities vectors from the word and the network embeddings improves the classification accuracy to 87% using random forest. This is a considerable improvement in comparison to the TF-IDF score which is 83% with all the features combined together. Furthermore, it has been noticed that classification with Table of Contents has a better micro F1 measure compared to the Named Entities with both Random Forest and CNN approach. Hence, it can be inferred that the word vectors for the words in the Table of Contents is capable of capturing more features relevant to the Wikipedia infobox type prediction problem compared to the entity vectors extracted from the RDFtoVec model. However, experiment with the CNN model by combining both word and entity embeddings together has not been performed because of the unequal length of the pre-trained vectors.

Last, Wikipedia articles in the ground truth i.e. the articles without infobox type, tend to be shorter and less informative due to various reasons such as, relatively new, less popular topic, lack of knowledge of the contributor on the topic etc. Therefore, the articles in the ground truth possess different characteristic as compared to the training data. Prediction of infobox types on the ground truth for the articles having TOC and abstract using the trained random forest model yields 53.7% micro-F1 score.

6 Conclusion

In this paper, a novel approach for Wikipedia infobox type prediction based on different types of embeddings has been analyzed. Also, the impact of using TOC

and Named Entities separately as features to predict an infobox type has been studied. The achieved results strengthen the fact that adding only TOC with the abstract in the feature set improves the accuracy of the classification process hugely. On the other hand, entities, if used together with the abstract, also have a positive impact on the classification process. Wikipedia infobox type prediction is an important task as KGs such as DBpedia are constructed by automatic information extraction from Wikipedia infoboxes. Hence, cleaning and assigning the correct types of Wikipedia infoboxes indirectly leads to an improvement of DBpedia type information. Additionally, this method can be extended to any number of Wikipedia infobox type classes with articles having abstract and/or TOC. Next, we would like to train CNN model combining both word and network embeddings. Moreover, as network embedding models can be applied in this task, we would like to train a network embedding model directly to the Wikipedia articles, instead of using Google vectors and analyze its impact in the classification process.

References

1. Bhuiyan, H., Oh, K., Hong, M., Jo, G.: An Unsupervised Approach for Identifying the Infobox Template of Wikipedia Article. In: CSE. pp. 334–338. IEEE Computer Society (2015)
2. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings. pp. 137–142 (1998)
3. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: EMNLP. pp. 1746–1751. ACL (2014)
4. Kliegr, T., Zamazal, O.: LHD 2.0: A Text Mining Approach to Typing Entities in Knowledge Graphs. J. Web Sem. 39, 47–61 (2016)
5. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: DBpedia–A large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web 6(2), 167–195 (2015)
6. Melo, A., Paulheim, H., Völker, J.: Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification. In: WIMS. p. 14 (2016)
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781 (2013)
8. Paulheim, H., Bizer, C.: Type Inference on Noisy RDF Data. In: ISWC. pp. 510–525 (2013)
9. Ristoski, P., Paulheim, H.: RDF2Vec: RDF Graph Embeddings for Data Mining. In: International Semantic Web Conference. pp. 498–514. Springer (2016)
10. Sultana, A., Hasan, Q.M., Biswas, A.K., Das, S., Rahman, H., Ding, C.H.Q., Li, C.: Infobox Suggestion for Wikipedia Entities. In: CIKM. pp. 2307–2310. ACM (2012)
11. Wu, F., Weld, D.S.: Autonomously Semantifying Wikipedia. In: CIKM. pp. 41–50. ACM (2007)
12. Yus, R., Mulwad, V., Finin, T., Mena, E.: Infoboxer: Using Statistical and Semantic Knowledge to Help Create Wikipedia Infoboxes. In: ISWC (Posters & Demos). CEUR Workshop Proceedings, vol. 1272, pp. 405–408. CEUR-WS.org (2014)