Learning Sequence Patterns in Knowledge Graph Triples to Predict Inconsistencies

Mahmoud Elbattah[1,2] and Conor Ryan[1]
[1]Department of Computer Science and Information Systems, University of Limerick, Ireland
[2]Laboratoire MIS, Université de Picardie Jules Verne, France

Keywords: Knowledge Graphs, Knowledgebase, Semantic Web, Machine Learning.

Abstract: The current trend towards the Semantic Web and Linked Data has resulted in an unprecedented volume of data being continuously published on the Linked Open Data (LOD) cloud. Massive Knowledge Graphs (KGs) are increasingly constructed and enriched based on large amounts of unstructured data. However, the data quality of KGs can still suffer from a variety of inconsistencies, misinterpretations or incomplete information as well. This study investigates the feasibility of utilising the subject-predicate-object (SPO) structure of KG triples to detect possible inconsistencies. The key idea is hinged on using the Freebase-defined entity types for extracting the unique SPO patterns in the KG. Using Machine learning, the problem of predicting inconsistencies could be approached as a sequence classification task. The approach applicability was experimented using a subset of the Freebase KG, which included about 6M triples. The experiments proved promising results using Convnet and LSTM models for detecting inconsistent sequences.

1 INTRODUCTION

The vision of the Semantic Web is to allow for storing, publishing, and querying knowledge in a semantically structured form (Berners-Lee, Hendler, and Lassila, 2001). To this end, a diversity of technologies has been developed, which contributed to facilitating the processing and integration of data on the Web. Knowledge graphs (KGs) have come into prominence particularly as one of the key instruments to realise the Semantic Web.

KGs can be loosely defined as large networks of entities, their semantic types, properties, and relationships connecting entities (Kroetsch and Weikum, 2015). At its inception, the Semantic Web has promoted such a graph-based representation of knowledge through the Resource Description Framework (RDF) standards. The concept was reinforced by Google in 2012, which utilised a vast KG to process its web queries (Singhal, 2012). The use of KG empowered rich semantics that could yield a significant improvement in search results.

Other major companies (e.g. Facebook, Microsoft) pursued the same path and created their own KGs to enable semantic queries and smarter delivery of data. For instance, Facebook provides a KG that can inspect semantic relations among entities (e.g. persons, places), which are all inter-linked in a huge social graph.

Equally important, many large-scale KGs have been made available thanks to the movement of Linked Open Data (LOD) (Bizer, Heath, and Berners-Lee, 2011). Examples include the DBpedia KG, which consists of about 1.5B facts describing more than 10M entities (Lehmann et al. 2015). Further openly available KGs were introduced over past years including Freebase, YAGO, Wikidata, and others. As such, the use of KGs has now become a mainstream for knowledge representation on the Web.

However, the data quality of KGs remains an issue of ongoing investigation. KGs are largely constructed by extracting contents using web scarpers, or through crowdsourcing. Therefore, the extracted knowledge could unavoidably contain inconsistencies, misinterpretations, or incomplete information. Moreover, data sources may include conflicting data for the same exact entity. For example, (Yasseri et al., 2014) analysed the top controversial topics in 10 different language versions of Wikipedia. Such controversial entities can lead to inconsistencies in KGs as well.

In this respect, this study explores a Machine Learning-based approach to detect possible

Elbattah, M. and Ryan, C.
Learning Triple Sequence Patterns in Knowledge Graphs to Predict Inconsistencies.
DOI: 10.5220/0008494604350441
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 435-441
ISBN: 978-989-758-382-7

inconsistencies within KGs. In particular, we investigated the feasibility of learning the sequence patterns of triples in terms of subject-predicate-object (SPO). The approach presented deemed successful for classifying triples including inconsistent SPO patterns. Our experiments were conducted using a large subset of Freebase data that comprised about 6M triples.

2 BACKGROUND AND RELATED WORK

This section reviews the literature from a two-fold perspective. Initially, the first part aims to get the reader's acquaintance with the quality aspects of KGs. The review particularly explores how the consistency of KGs was described in literature. Subsequently, we present selective studies that introduced methods to deal with consistency-related issues in KGs.

2.1 Quality Metrics of KGs

With ongoing initiatives for publishing data into the Linked Data cloud, there has been a growing interest in assessing the quality of KGs. Part of the efforts was basically directed towards discussing the different quality aspects of KGs (e.g. Zaveri et al., 2016; Debattista et al. 2018; Färber et al. 2018). This section briefly reviews representative studies in this regard. Particular attention is placed on the consistency of KGs, which is the focus of this study.

(Zaveri et al. 2016) defined three categories of KG quality including: i) Accuracy, ii) Consistency, and iii) Conciseness. The consistency category contained several metrics such as misplaced classes or properties, or misuse of predicates. Likewise, (Färber et al., 2018) attempted to define a comprehensive set of dimensions and criteria to evaluate the data quality of KGs. The quality dimensions were organised into four broad categories as follows: i) Intrinsic, ii) Contextual, iii) Representational data quality, and iv) Accessibility. The consistency of KGs was included under the *Intrinsic Category*. In particular, consistency-related criteria were given as follows:

- Check of schema restrictions during inserting new statements.
- Consistency of statements against class constraints.
- Consistency of statements against relation constraints.

2.2 Refinement of KGs

Various methods were proposed for the validation and refinement of the quality of KGs. In general, there have been two main goals of KG refinement including (Paulheim, 2017): i) Adding missing knowledge to the graph, and ii) Identifying wrong information in the graph. The review here is more focused on studies that addressed the latter case.

One of the early efforts was the DeFacto (Deep Fact Validation) method (Gerber et al., 2015). The DeFacto approach was based on finding trustworthy sources on the Web in order to validate KG triples. This could be achieved by collecting and combining evidence from webpages in several languages. In the same manner, (Liu, d'Aquin, and Motta, 2017) presented an approach for the automatic validation of KG triples. Named as Triples Accuracy Assessment (TAA), their approach works by finding a consensus of matched triples from other KGs. A confidence score can be calculated to indicate the correctness of source triples.

Other studies attempted to use Machine Learning to detect the quality deficiencies in KGs. In this regard, association rule mining has been considered as a suitable technique for discovering frequent patterns within RDF triples. For instance, (Paulheim, 2012) used rule mining to find common patterns between types, which can be applied to knowledgebases. The approach was experimented on DBpedia providing results at an accuracy of 85.6%. Likewise, (Abedjan. and Naumann, 2013) proposed a rule-based approach for improving the quality of RDF datasets. Their approach can help avoid inconsistencies through providing predicate suggestions, and enrichment with missing facts. Further studies continued to develop similar rule-based approaches such as (Barati, Bai, and Liu, 2016), and others.

More recently, (Rico et al., 2018) developed a classification model for predicting incorrect mappings in DBpedia. Different classifiers were experimented, and the highest accuracy (≈ 93%) could be achieved using a Random Forest Model.

However, further potentials for using Machine Learning may have not been explored in literature yet. For instance, utilising the sequence-based nature of triples for learning, which is the focus of this study. Our approach is based on applying sequence classification techniques to the SPO triples in KGs. To the best of our knowledge, such approach has not been applied before.

3 DATA DESCRIPTION

As mentioned earlier, the study's approach was experimented on the Freebase KG. The following sections describe the dataset used along with a brief review of the key features of Freebase.

3.1 Overview

Freebase was introduced as a huge knowledgebase of cross-linked datasets. Freebase was initially launched by Metaweb Technologies before it was acquired by Google. The knowledgebase was constructed using a wide diversity of structured data imported from various sources such as Wikipedia, MusicBrainz and WordNet (Bollacker, Cook, and Tufts, 2007).

In 2016, the Freebase KG has been merged with Wikidata to form even a larger KG (Pellissier Tanon et al., 2016). However, Freebase data is still freely accessible through a downloadable dump (≈250 GB). In this study, we used the reduced version (≈5GB), which contain the basic identifying facts.

The reduced subset contained more than 20M triples.

For further simplification, our dataset included 6M triples only. The triples spanned a variety of domains, which are organised into broad categories as: i) Science & Tech, ii) Arts & Entertainment, iii) Sports, iv) Society, v) Products & Services, vi) Transportation, vii) Time & Space, viii) Special Interests, and ix) Commons. Table 1 gives summary statistics of the dataset with respect to each category.

Table 1: Statistics of the dataset.

Freebase Category	Count of Triples
#1 Arts & Entertainment	2,461,499
#2 Time & Space	1,483,723
#3 Society	658,823
#4 Science & Tech.	578,895
#5 Products & Services	485,406
#6 Special Interests	178,605
#7 Transportation	116,071
#8 Sports	36,978

3.2 Freebase Knowledge Graph

Like the RDF model, the knowledge in Freebase is represented as SPO triples, which collectively form a huge KG. The KG contains millions of topics about real-world entities including people, places and things. A topic may refer to a physical entity (e.g. Albert Einstein) or an abstract concept (e.g. Theory of Relativity). Figure 1 illustrates a basic example of triples in Freebase KG.

Furthermore, Freebase was thoroughly architected around a well-structured schema. Entities in the KG are mapped to abstract *types*. A type serves as a conceptual container of properties commonly needed for characterising a particular class of entities (e.g. Author, Book, Location etc.). In this manner, entities can be considered as instances of the schema types.

The Freebase KG incorporates around 125M entities, 4K types, and 7K predicates (Bollacker et al. 2008). Entities can be associated with any number of types. As an example, Figure 2 demonstrates the multi-typing of the famed British politician, *Winston Churchill*. As it appears, Churchill can be described as a type of *Politician*, *Commander*, *Author*, *Nobel Laureate*, *Visual Artist*, and generally as a type of *Person*. This presents a good example of how the multi-faceted nature of real-world entities is represented in the KG.

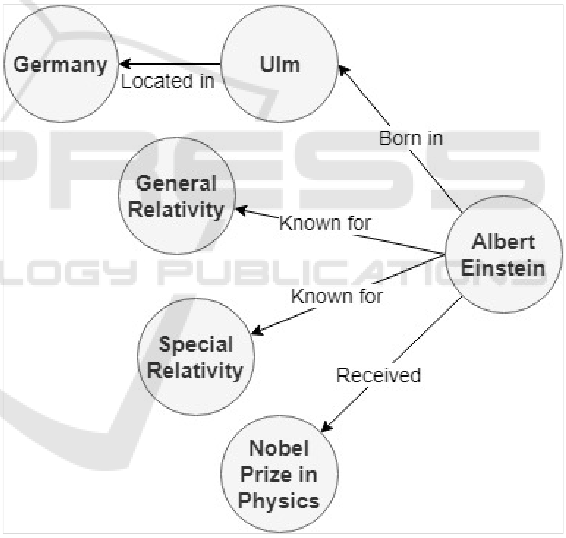


Figure 1: Representation of knowledge in Freebase.

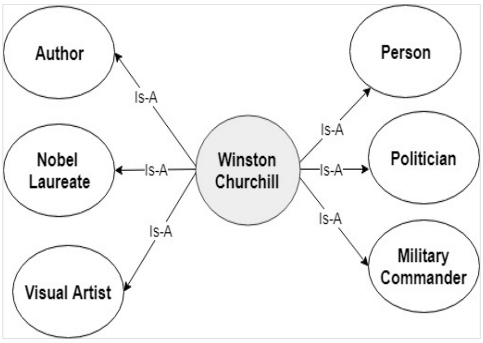


Figure 2: Example of entity types in Freebase.

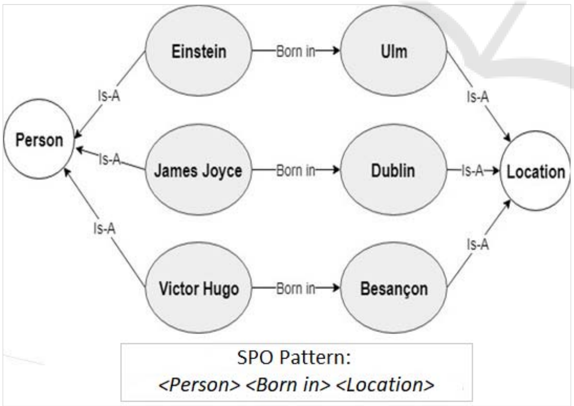
4 OUR APPROACH

The study was motivated by developing an approach to help predict possible inconsistencies in KGs. In essence, our approach is based on the premise that the sequence-based patterns of SPO triples can discriminate consistent statements against others. Using Machine learning, the problem could be approached as a binary classification task. The following sections elaborates our approach and the underlying key ideas.

4.1 Key Idea I: Extracting Unique SPO Patterns in Knowledge Graph

The first challenge was to generalise the SPO patterns existing in a huge KG. With millions of triples, the learning process becomes very computationally expensive, and prone to overfitting as well.

The key idea was to avail of the generic entity types defined by Freebase in order to provide a higher-level abstraction of the SPO triples. The subject and object in each triple were mapped to one or more of the entity types (e.g. Person, Author) as explained before. For example, Figure 3 presents a set of triples that include different subjects and objects. However, they all can be conceptualised as a generic sequence in terms of: <Person> <Born in> <Location>. This can be the case for thousands or millions of triples that represent the same sequence.



SPO Pattern:
<Person> <Born in> <Location>

Figure 3: Example of extracting unique SPO patterns.

As such, the generic SPO statements can describe the common behavioural patterns found in the KG, which are likely to be consistent. Further, the problem dimensionality could be significantly reduced by decreasing the number of sequences under consideration. Specifically, the dataset initially contained 6M triples, while the unique patterns were only about 124K. This contributed to making the problem more amenable for Machine Learning.

4.2 Key Idea II: Generating Syntenic False Patterns

One major limitation to developing a Machine Learning-based approach was the unavailability of wrong (i.e. inconsistent) examples. The triples included in the KG were presumably considered as correct. Even though inconsistent examples can likely exist, they were not explicitly labelled.

In this regard, the second key idea was to generate SPO patterns that can serve as synthetic samples of inconsistent sequences. The generated patterns were checked such that they did not exist within the set of true patterns. For instance, a generated pattern could be like: <Location> <Born in> <Person>.

A Long Short-Term Memory (LSTM) model was used to generate the synthetic sequences. LSTM models can perform as predictive and generative models as well. They can learn data sequences (e.g. time series, texts, audio), and then generate entirely new plausible sequences. The LSTM was proposed by (Hochreiter and Schmidhuber, 1997), which proved very successful for tackling sequence-based problems (e.g. Graves, Mohamed, and Hinton, 2013; Oord et al., 2016; Gers, Eck, Schmidhuber, 2002). Instead of neurons, LSTM networks include memory cells, which have further components to deal with sequence inputs. A cell can learn to recognise an input, store it in the long-term state. The input can be preserved and extracted whenever needed. (Géron, 2017) would be a good resource for a detailed explanation of the LSTM mechanism, as this should go beyond the scope and space of the study.

5 EXPERIMENTS

5.1 Computing Environment

We used the Data Science Virtual Machine (DSVM) provided by the Azure platform. The DSVM greatly facilitates compute-intensive tasks using GPU-optimized VMs. The DSVMs are powered by the NVIDIA Tesla K80 card and the Intel Xeon E5-2690 v3 processor.

In our case, the DSVM included double GPUs and 12 vCPUs with 112 GB RAM. All models were trained using the same DSVM settings.

5.2 Data Preprocessing

A set of preprocessing procedures were applied in order to make the sequences suitable for learning. Initially, the first step was to transform the raw text sequences into tokens or words. That process is called *tokenisation*.

The Keras library (Chollet, 2015) provides a convenient method for text tokenisation, which we used for preprocessing the sequences. Using the *Tokenizer* utility class, textual sequences could be vectorised into a list of integer values. Each integer was mapped to a value in a dictionary that encoded the entire corpus, where keys in the dictionary representing the vocabulary terms themselves.

The second step was to represent tokens as vectors, by applying the so-called *one-hot encoding*. This is a simple process that produces a vector of the length of the vocabulary with an entry (i.e. one) for each word in the corpus. In this way, each word would be given a spot in the vocabulary, where the corresponding index is set to one. Keras also provides easy-to-use APIs for applying the one-hot encoding.

The final step was to use *word embedding*, which is a vital procedure for making the sequences amenable for Machine Learning. The one-hot encoded vectors are very high-dimensional and sparse. Embeddings are used to provide dense word vectors of much lower dimensions of the encoded representations.

Keras provides a special layer for the implementation of word embeddings. The embedding layer can be conceived as a dictionary that maps integer indices into dense vectors (Chollet, 2017). It takes integers as input, then it looks up these integers in an internal dictionary, and it returns the associated vectors. The embedding layer was used as the top layer in the generative and classification models.

5.3 Generative Model

The generative model is a typical LSTM implantation, which comprised a single layer of 100 cells. The model was implemented using the Keras *CuDNNLSTM* layer, which includes optimized routines for GPU computation.

Figure 4 demonstrates the model loss in training and validation over 20 epochs with 30% of the dataset used for validation. Training the model took ≈17 minutes using the double-GPU VM. Eventually, the model was used to generate about 98K new sequences that did not exist in the original KG. The implementation of the model is shared on our GitHub repository (Elbattah, 2019).

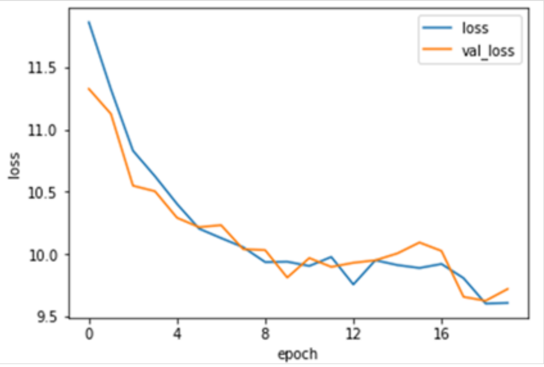


Figure 4: Generative model loss in training and validation sets.

5.4 Classification Model

The final dataset contained more than 222K sequences including the original triples along with the synthetic samples. The classification model was trained using LSTM and ConvNet models. The architectures of the ConvNet and LSTM models are given in Figure 5 and Figure 6 respectively. Both models were trained using the same set of hyperparameters (e.g. epochs=10, batch size=128).

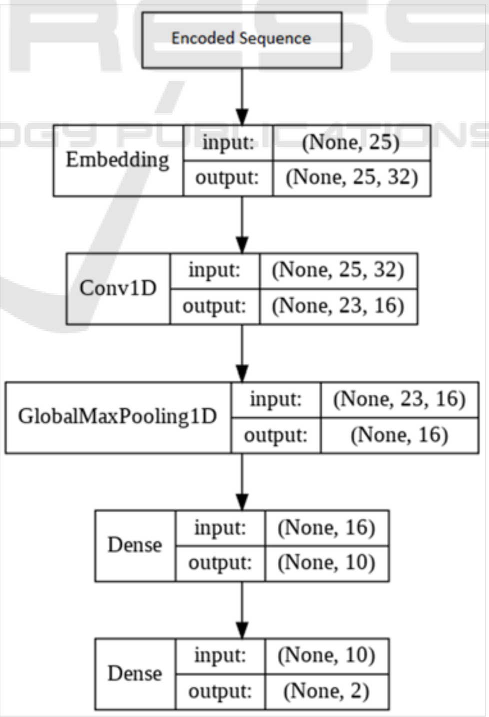


Figure 5: Architecture of the ConvNet model.

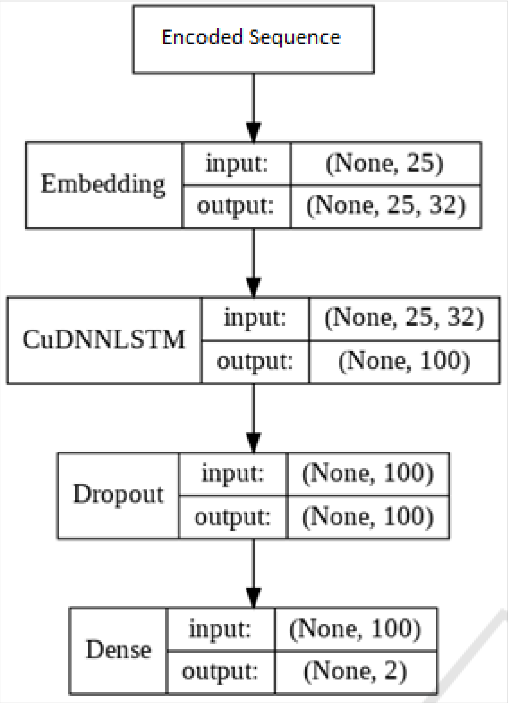


Figure 6: Architecture of the LSTM model.

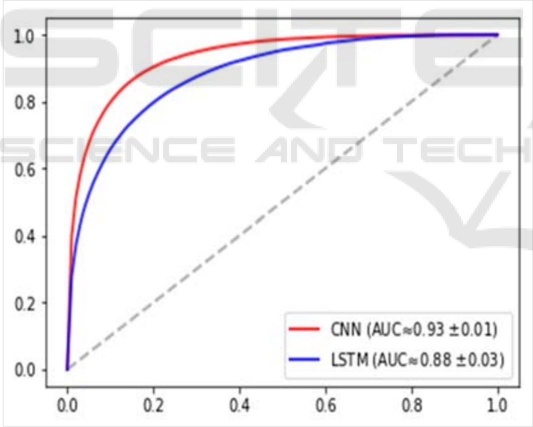


Figure 7: ROC curves of the classification models.

The classification accuracy was analysed based on the Receiver Operating Characteristics (ROC) curve. The ROC curve plots the relationship between the true positive rate and the false positive rate across a full range of possible thresholds. Figure 7 plots the ROC curves for the ConvNet and LSTM models. The figure also shows the approximate value of the area under the curve and its standard deviation over the 3-fold cross-validation. At it appears, the ConvNet model could achieve the highest accuracy (≈93.0%). The implementations of both models are shared as Jupyter Notebboks on our GitHub repository (Elbattah, 2019).

6 CONCLUSIONS

The sequence-based representation of SPO triples can serve as a basis for predicting inconsistencies in KGs. The study presented the idea of utilising Freebase-defined types to extract the unique SPO patterns in the KG. Using Machine Learning, the problem of detecting inconsistencies could be approached as a sequence classification task. The validity of the method was experimented using a subset of the Freebase KG. The SPO-based sequences were used to train a binary classification. High accuracy could be achieved using ConvNet and LSTM models. However, one key limitation of this work is that the inconsistent patterns were based on synthetic samples produced by a generative model.

REFERENCES

Abedjan, Z. and Naumann, F., 2013. Improving RDF data through association rule mining. Datenbank-Spektrum, 13(2), pp.111-120.

Barati, M., Bai, Q. and Liu, Q., 2016. SWARM: an approach for mining semantic association rules from semantic web data. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence (pp. 30-43). Springer, Cham.

Berners-Lee, T., Hendler, J. and Lassila, O., 2001. The semantic web. Scientific American, 284(5), pp.28-37.

Bizer, C., Heath, T. and Berners-Lee, T., 2011. Linked data: The story so far. In Semantic Services, Interoperability and Web Applications: Emerging Concepts (pp. 205-227). IGI Global.

Bollacker, K., Cook, R. and Tufts, P., 2007. Freebase: A shared database of structured general human knowledge. *In Proceedings of the 22nd national Conference on Artificial intelligence (AAAI)* - Volume 2, 207.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J., 2008, June. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1247-1250). ACM.

Chollet, F., 2015. Keras. https://github.com/fchollet/keras

Chollet, F., 2017. Deep learning with Python. Manning Publications.

Debattista, J., Lange, C., Auer, S. and Cortis, D., 2018. Evaluating the quality of the LOD cloud: An empirical investigation. *Semantic Web*, , pp.1-43. IOS Press.

Elbattah, M., 2019. https://github.com/Mahmoud-Elbattah/Learning_Triple_Sequences_in_Knowledge_Graphs

Färber, M., Bartscherer, F., Menne, C. and Rettinger, A., 2018. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1), pp.77-129. IOS Press

Gerber, D., Esteves, D., Lehmann, J., Bühmann, L., Usbeck, R., Ngomo, A.C.N. and Speck, R., 2015. Defacto—temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35, pp.85-101. Elsevier.

Géron, A., 2017. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. pp.401-405. *O'Reilly Media, Inc*.

Gers, F.A., Eck, D. and Schmidhuber, J., 2002. Applying LSTM to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri*-01 (pp. 193-200). Springer.

Graves, A., Mohamed, A.R. and Hinton, G., 2013, May. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE international Conference on Acoustics, Speech and Signal Processing (pp. 6645-6649). IEEE.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, 9(8), pp.1735-1780. MIT Press.

Kroetsch, M. and Weikum, G., 2015. Special issue on knowledge graphs. Journal of Web Semantics. Elsevier.

Liu, S., d'Aquin, M. and Motta, E., 2017. Measuring accuracy of triples in knowledge graphs. In Proceedings of the International Conference on Language, Data and Knowledge (pp. 343-357). Springer, Cham.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S. and Bizer, C., 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2), pp.167-195. IOS Press.

Oord, A.V.D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

Paulheim, H., 2012. Browsing linked open data with auto complete. In Proceedings of the Semantic Web Challenge, co-located with the 2012 ISWC Conference, Springer.

Paulheim, H., 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3), pp.489-508. IOS Press.

Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T. and Pintscher, L., 2016, April. From freebase to wikidata: The great migration*. In Proceedings of the 25th International Conference on World Wide Web* (pp. 1419-1428).

Rico, M., Mihindukulasooriya, N., Kontokostas, D., Paulheim, H., Hellmann, S. and Gómez-Pérez, A., 2018, April. Predicting incorrect mappings: a data-driven approach applied to DBpedia. *In Proceedings of the 33rd Annual ACM Symposium on Applied Computing* (pp. 323-330). ACM.

Singhal, A. Introducing the knowledge graph: things, not strings, May 16, 2012. URL: https://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html.

Yasseri, T., Spoerri, A., Graham, M. and Kertész, J., 2014. The most controversial Topics in Wikipedia. *Global Wikipedia: International and Cross-Cultural Issues in Online Collaboration* (25).

Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J. and Auer, S., 2016. Quality assessment for linked data: A survey. *Semantic Web*, 7(1), pp.63-93. IOS Press.