Refinement of the Wikidata taxonomy with neural networks

Proposal for Bachelor thesis

Alex Baier
abaier@uni-koblenz.de

December 1, 2016

1 Motivation

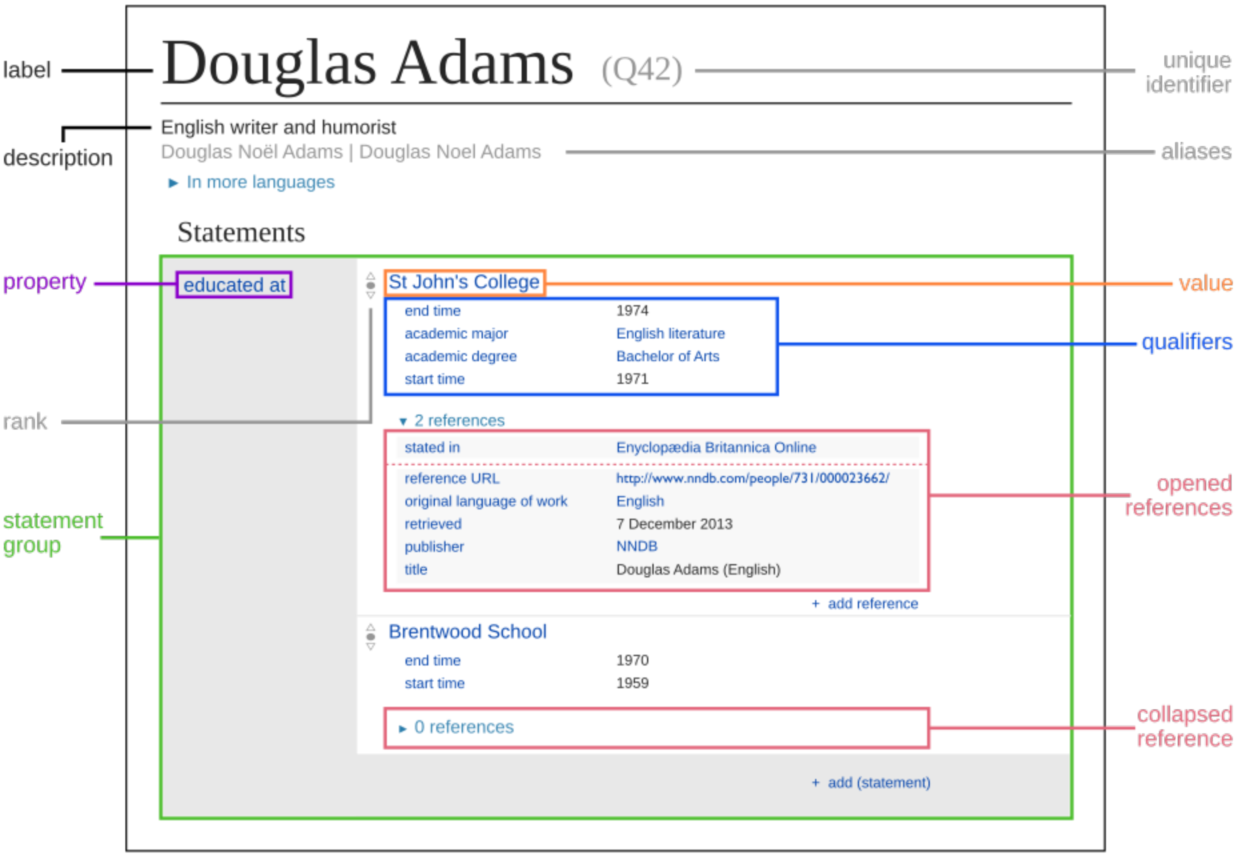
Wikidata is an open, free, multilingual and collaborative knowledge base. It acts as a structured knowledge source for other Wikimedia projects. It tries to model the real world, meaning every concept, object, animal, person,etc., therefore Wikidata can always be considered to be incomplete. Wikidata is mostly edited and extended by humans, which implies entries in Wikidata can be erroneous. On 7th November 2016 it contained 24,438,781 entities [2].

Most entities in Wikidata are items. Items consist of labels, aliases and descriptions in different languages. Sitelinks connect items to their corresponding Wiki articles. Most importantly items are described by statements. Statements are in their simplest form a pair of property and value. They can be annotated with references and qualifiers. See figure 1 for an example.

The other category of entities in Wikidata are properties. Properties are used to describe data values of items. A property always has a data type, which are for example item or date. Two important properties are *instance of (P31)* and *subclass of (P279)*. The data type of both properties are item, which means they are used to connect two items with a subclass or instance relationship.

The *subclass of (P279)* property allows the creation of a taxonomy in Wikidata. Figure 2 shows a fragment of Wikidata's taxonomy. It can, for example, be seen that *electrical apparatus (Q2425052)* is the superclass of *Computer (Q68)*, *clock (Q376)*, and 4 other classes. Taxonomies like this can be used for different tasks. [12] for example develops a method of word classification in thesauri, which exploits the structure of taxonomies. Other uses may be found in information retrieval and reasoning.

As of the 7th November 2016 over a million classes are present in this taxonomy. A root class in a taxonomy is a class, which has no more generalizations. Root classes should therefore describe the most basic concepts. For example *entity (Q35120)* can be considered to most general class, comparable to the Object class in Java. According to



Douglas Adams (Q42)

English writer and humorist

Douglas Noël Adams | Douglas Noel Adams

► In more languages

Statements

educated at

St John's College

end time	1974
academic major	English literature
academic degree	Bachelor of Arts
start time	1971

▼ 2 references

stated in	Encyclopædia Britannica Online
reference URL	http://www.nndb.com/people/731/000023662/
original language of work	English
retrieved	7 December 2013
publisher	NNDB
title	Douglas Adams (English)

+ add reference

Brentwood School

end time	1970
start time	1959

► 0 references

+ add (statement)

Figure 1: Graphic representing the datamodel in Wikidata with a statement group and opened reference; `https://commons.wikimedia.org/wiki/File:Graphic_representing_the_datamodel_in_Wikidata_with_a_statement_group_and_opened_reference.svg`

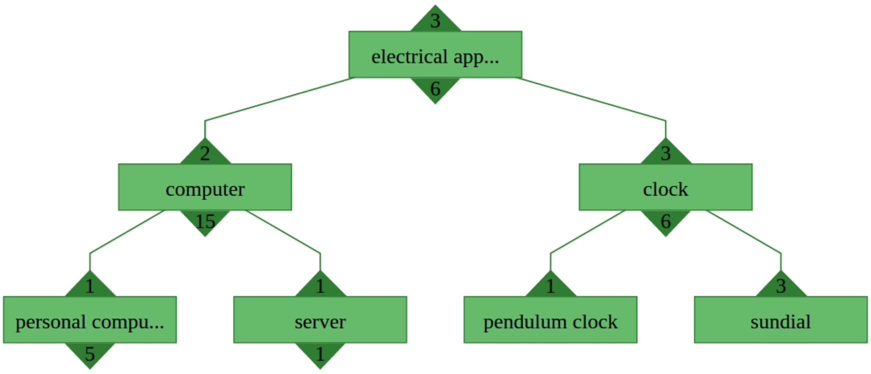


Figure 2: Fragment of Wikidata's taxonomy created with [15]

this view, we would assume that a good taxonomy has only very few, possibly only one root class.
At the current state (2016-11-07) Wikidata contains 7142 root classes, of which 5332 have an English label. There are many root classes for which we easily can find generalizations. For example both *Men's Junior European Volleyball Championship (Q169359)* and *Women's Junior European Volleyball Championship (Q169956)* are root classes. By just looking at their labels we can find an appropriate superclass, *European Volleyball Championship (Q6834)*, for both of them. A superclass should be considered appropriate, if it is a generalization of the child class and also the most similar respectively nearest class to the child class. Tools, which solve this task, may help the Wikidata community in improving the existing taxonomy. Similar problems in the field of ontology and taxonomy learning are already well researched (see section "Related Work"). However neural networks are comparably scarcely applied in this research area. Neural networks have proven to be very powerful for other complex tasks, e.g. speech recognition [11]. Accordingly it may be interesting to see, how neural networks can be used for the task of refining the taxonomies of knowledge bases by reducing the number of root classes.

2 Problem statement

To define the problem following definitions are needed:

Definition 1 (Directed Graph)**.** A graph G is an ordered pair $G = (V, E)$, where V is a set of vertices, and $E = \{(v_1, v_2) \mid v_1, v_2 \in V\}$ is a set of ordered pairs called directed edges, connecting the the vertices.

Definition 2 (Predecessor, Successor)**.** Let $G = (V, E)$ be a directed graph.
$v_1 \in V$ is a predecessor of $v_2 \in V$, if there exists an edge so that $(v_1, v_2) \in E$.
Let $v \in V$ be a vertice of G, then $pred(v) = \{w \mid (w, v) \in E\}$ is the set of predecessors of v.
$v_1 \in V$ is a successor of $v_2 \in V$, if there exists an edge so that $(v_2, v_1) \in E$.
Let $v \in V$ be a vertice of G, then $succ(v) = \{w \mid (v, w) \in E\}$ is the set of successors of v.

Definition 3 (Walk)**.** Let $G = (V, E)$ be a directed graph.
A walk W of length $n \in \mathbb{N}$ is a sequence of vertices $W = (v_1, \ldots, v_n)$ with $v_1, \ldots, v_n \in V$, so that $(v_i, v_{i+1}) \in E \; \forall i = 1, \ldots, n-1$.

Definition 4 (Cycle)**.** A walk $W = (v_1, \ldots, v_n)$ of length n is called a cycle, if $v_1 = v_n$.

Definition 5 (Path)**.** A walk $P = (v_1, \ldots, v_n)$ is a path from v_1 to v_n, if $v_i \neq v_j$ for all $i, j = 1, \ldots, n$ with $i \neq j$.

Definition 6 (Acyclic Graph)**.** A directed graph G is called acyclic graph, if there are no cycles in G.

Definition 7 (Statement)**.** **TODO 1: Define statement.**

Definition 8 (Class)**.** A class is a tuple $(id, label, Statements, Instances, wiki)$:

- $id \in \mathbb{N}$, which is a numerical Wikidata item ID;
- $label$, which is the, to id corresponding, English label in Wikidata;
- $Statements$ is a set of statements about the class;
- $Instances \in \mathcal{P}(\mathbb{N})$ is the set of numerical Wikidata item IDs, which are instances of the class;
- $wiki$ is the, to the class corresponding, English Wikipedia article text.

Definition 9 (Taxonomy)**.** A taxonomy $T = (C, S)$ is a acyclic graph, where C is a set of classes, and S is a set of subclass-of relations between these classes.

Definition 10 (Subclass Relation)**.** Let $T = (C, S)$ be a taxonomy.
The transitive, ordered relation $\lhd_{subclass}$ is defined.
Let $c_1, c_2 \in C$. $c_1 \lhd_{subclass} c_2$, if there is a path $P = (c_1, \ldots, c_2)$ from c_1 to c_2 in T.

Definition 11 (Root class)**.** Let $T = (C, S)$ be a taxonomy.
$r \in C$ is called root class of T, if $|succ(r)| = 0$.
$root(T) = \{r \in C \mid |succ(r)| = 0\}$ is the set of all root classes in T.

Finally we can define our problem as the following task:

Problem. Let W_1 be the taxonomy of Wikidata, where only labeled root classes are considered. On 7th November 2016 the following state applies $|root(W_1)| = 5332$.
$W_1 = (C, S)$ is the input for the described problem.
Let W_2 be the refined output taxonomy.
A refinement method is needed to significantly reduce the number of root classes in the Wikidata taxonomy. After the refinement method is applied on W_1, which outputs W_2, the following should be true: $|root(W_2)| \ll |root(W_1)|$.

The refinement process can be reduced to the following smaller task:
Let $r \in root(W_1)$.
Find a $c \in C$ with $\neg(c \lhd_{subclass} r)$, so that c is the most similar super class of r.
Connecting r to c with an edge produces the output taxonomy $W_2 = (C, S \cup \{(r, c)\})$.
Accordingly $|root(W_2)| = |root(W_1)| - 1$ applies.
Repeating this smaller task will eventually yield $|W_2| \ll |W_1|$.

The problem can therefore be defined as developing a method, which finds, given a taxonomy $W = (C, S)$ and a root class $r = root(W)$, the most similar superclass of r.

TODO 2: Define similarity between classes. How? Many aspects to consider.
TODO 3: Probably need to define neural networks.

3 Related work

The research fields of ontology learning and neural networks are of interest to the proposed thesis.

The topic of ontology learning is defined and analyzed in [9] by Maedche and Staab. Additionally a tool called *OntoEdit* was developed in the process. [9] considers a semi-automatic approach and divides the process of ontology learning into the following steps: **import/reuse** existing ontologies, **extract** major parts of target ontology, **prune** to adjust the ontology for its primary purpose, **refine** ontology to complete it at a fine granularity, and **apply** it on target application to validate the results.
The problem solved by this thesis belongs to the step **refine**. Even though different approaches for refinement are considered and implemented in the paper, the use of neural networks is not considered.

[12] by Pekar & Staab define algorithms for classification, which exploit the structure of taxonomies. Distributional and taxonomic similarity measures used on nearest neighborsare used to make a classification decision. These algorithms are applied on the classification of new words (instances) into thesauri. In comparison the target of this thesis will be to improve the existing taxonomy. For this the closest generalizations of root classes have to be found. The similarity measures used in [12] may prove useful for the defined problem.

Both [4] and [14] develop neural networks, deep neural network and recursive neuron network, which are able to encode graphs as vectors. It is proposed by both papers to use the generated vectors as input for classification methods. Because the networks are defined in such a way that semantic information of the graph is preserved to some degree, the vectors could be used for other task like measuring the similarity of classes based on their position in the taxonomy using for example cosine similarity.

[10] by Mikolov et al. defines two neural network models, Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram, which are able to create word vector representations. They capture the semantics of words very well and preserve linear regularities between words. [12] uses word representations based on counts with context words. It is possible that the use of newer word representation model like CBOW will also improve their classification method.

A recurrent neural network model for ontology learning is described in [13] by Petrucci, Ghidini & Rospocher. Using encyclopedic text as input OWL formulas are created. The authors argue that their model should be effective, because neural networks have shown success in natural language processing tasks. At this time the described model is under evaluation, so it is not shown that the model will generate good results. Different subtasks of ontology learning are solved by the paper and by the proposed thesis. Additionally the thesis will contain an evaluation of the created model and it can be shown if neural networks are a sensible method for ontology learning.

In conclusion, for the task of ontology learning, especially the task of taxonomy refinement, different methods already exist and have proven to work well. But most of these methods do not consider the use of neural networks. In the proposed thesis, a solution for taxonomy refinement in a specific context will be developed.

4 Methodology

For solving the defined problem with a neural network, the following methodology is proposed:

First the current taxonomy of Wikidata needs to be analyzed. I should be answered, how many classes and especially root classes are available and what their characteristics are, e.g. number of instances and subclasses, how many and what kind of statements. This will allow a more focused search and analysis in the following step.

The literature about neural networks needs be researched. Different neural network models will be analyzed and compared, regarding input, output, task and performance. Furthermore the neural networks should be compared to other solutions for the same tasks, so the decision to use neural networks can be motivated.

This leads to the development of a new neural network architecture based on the researched networks, which is specialized to solve the defined problem. The decision made in the development will be justified based on the results of the previous steps.

After the neural network is developed, the system needs to implemented and training data needs to be collected. The implemented network will be trained, and then tested. Reconfiguration of the network and modification of training data will be repeated, until the test results are satisfactory.

In the last step the neural network will be used on all identified root classes. An evaluation with the Wikidata community will be executed. In this evaluation participants will be asked to rate the results of the neural network. This will answer the question of how accurate the developed method is, and may identify problems, e.g. results are too general, a certain category of root classes could not be correctly classified at all. The evaluation results will be analyzed and possible improvements for the network discussed.

The evaluation with the Wikidata community is very important, because a tool based on the developed method should ideally be used by users to support the curation process in Wikidata. Therefore the community would need to agree with the results of the method, otherwise such a tool would serve no practical purpose.

5 Expected results

TODO 4: What should I expect? NNs are powerful, so it is likely to work, if the architecture is well designed and the chosen data is good.

6 Time plan

TODO 5: Add a time plan.

List of Figures

References

[1] Wikidata game. https://tools.wmflabs.org/wikidata-game/.

[2] Wikidata statistics. https://tools.wmflabs.org/wikidata-todo/stats.php.

[3] Ebru Arisoy, Tara N. Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Deep neural network language models. In Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT, WLM '12, pages 20–28, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In Dale Schuurmans and Michael P. Wellman, editors, AAAI, pages 1145–1152. AAAI Press, 2016.

[5] Claudia d'Amato, Steffen Staab, Andrea G. B. Tettamanzi, Tran Duc Minh, and Fabien L. Gandon. Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases. In Sascha Ossowski, editor, SAC, pages 333–338. ACM, 2016.

[6] Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra. DRAW: A recurrent neural network for image generation. CoRR, abs/1502.04623, 2015.

[7] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. CoRR, abs/1404.2188, 2014.

[8] Alexander Maedche, Viktor Pekar, and Steffen Staab. Ontology Learning Part One — on Discovering Taxonomic Relations from the Web, pages 301–319. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[9] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. IEEE Intelligent Systems, 16(2):72–79, March 2001.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[11] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010, pages 1045–1048, 2010.

[12] Viktor Pekar and Steffen Staab. Taxonomy learning - factoring the structure of a taxonomy into a semantic classification decision. In Proceedings of the 19th Conference on Computational Linguistics, COLING-2002, August 24 - September 1, 2002 , Taipei, Taiwan, 2002, 2002.

[13] Giulio Petrucci, Chiara Ghidini, and Marco Rospocher. Using recurrent neural network for learning expressive ontologies. CoRR, abs/1607.04110, 2016.

[14] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks, 8(3):714–735, may 1997.

[15] Serge Stratan. Wikidata taxonomy browser (beta). `http://sergestratan.bitbucket.org/`.

[16] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. ACM Comput. Surv., 44(4):20:1–20:36, September 2012.

[17] Guoqiang Peter Zhang. Neural networks for classification: a survey. In and Cybernetics - Part C: Applications and Reviews, 2000.