# Refinement of the Wikidata taxonomy with neural networks

**Proposal for Bachelor thesis**

Alex Baier
abaier@uni-koblenz.de

November 27, 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. [9] 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

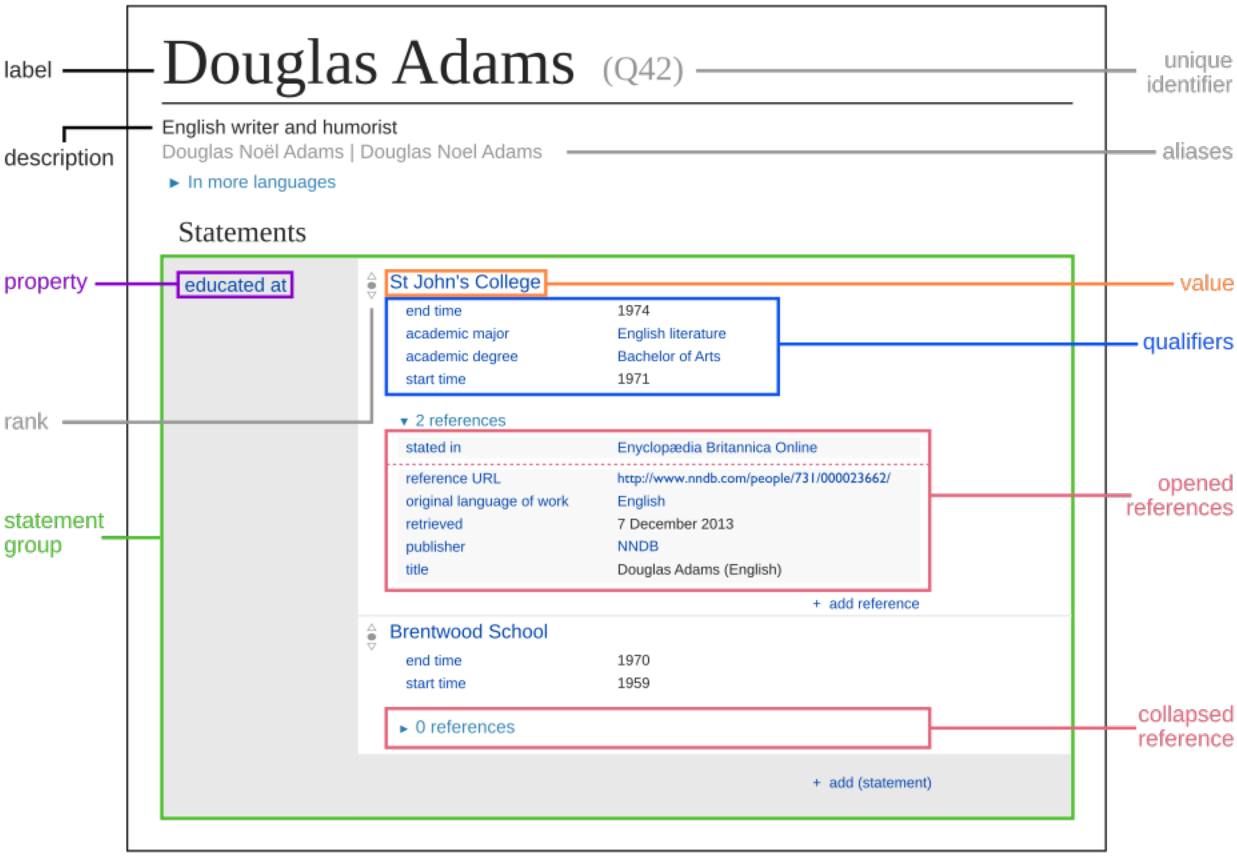


Figure 1: Graphic representing the datamodel in Wikidata with a statement group and opened reference; `https://commons.wikimedia.org/wiki/File:Graphic_representing_the_datamodel_in_Wikidata_with_a_statement_group_and_opened_reference.svg`

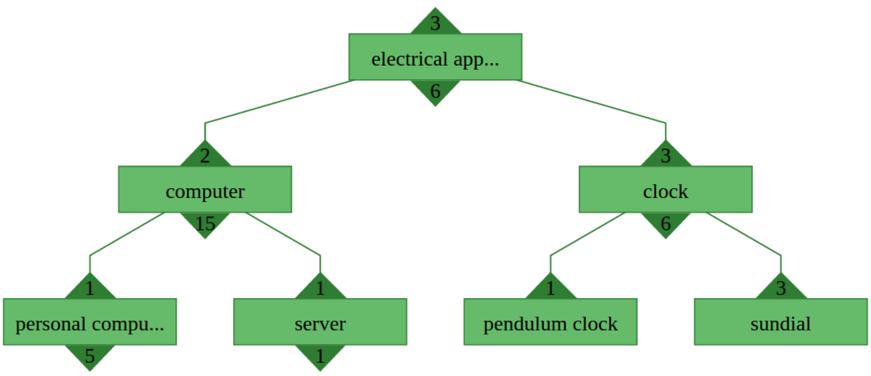


Figure 2: Fragment of Wikidata's taxonomy created with [12]

considered to most general class, comparable to the Object class in Java. According to 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 [8]. 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)**.**
**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 closest appropriate generalization 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 closest generalization of $r$.

**TODO 1: Define what generalization means, what closest means?**

## 3 Related work

**TODO 2: Add literature about neural networks, ontology learning.**

## 4 Methodology

**TODO 3: Order is not correct, rewrite this.** The main task of this work will be to develop a method for extending taxonomic relations of root classes based on neural networks. The task can be divided into the following subtasks:

The root classes in the knowledge base have to be identified and analyzed. This should result in a comparison of similarities and characteristics of root classes in Wikidata. The purpose of this task is to identify, which data is available and how it is structured.

Two categories of to this challenge related work has to analyzed. The first category of related work is concerned with the topic of ontology and taxonomy learning. The second category is concerned with different applications of neural networks, and ways to represent complex data for usage in neural networks. Goal of this task should be to find an appropiate mapping of Wikidata root classes to feature vectors, which can be used by neural networks.

After the mapping of classes is defined, data can be collected per hand or possibly by crawling for training neural networks. The author will create a ground truth for the collected data.

Finally different neural networks can be implemented and trained using existing libraries. The configurations of the networks will be improved by means of experimentation and literature review **TODO 4: Is "literature review" the right word?** until a small enough error in testing is achieved.

The real data, all 7142 root classes, will be applied on the best performing network(s). The results will be reviewed by the author. If it is possible a survey with the Wikidata community will be executed. A (random?) subset of the results will be presented to the community, and the participants of the survey will be asked, whether they think the generated suggestions of the network are accurate and could be entered into Wikidata. Such a survey would be really important to confirm the validity and relevance of this work.

## 5 Expected results

**TODO 5: What should I expect? NNs are powerful, so it is likely to work, if the right data is used.**

# 6 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] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

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

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

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

[13] 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.