Refinement of the Wikidata taxonomy with neural networks

Proposal for Bachelor thesis

Alex Baier
abaier@uni-koblenz.de

December 12, 2016

1 Motivation

Wikidata is an open, free, multilingual and collaborative knowledge base. It is as a structured knowledge source for other Wikimedia projects. It tries to model the real world, meaning every concept, object, animal, person,etc.. We call these entities notorious entities. Wikidata is mostly edited and extended by humans, which in general improves the quality of entries compared to fully-automated systems, because different editors can validate and correct occurring errors.
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.

An important property used to describe a Wikidata item is *subclass of (P279)*. Items, which contain statements with this property, are classes, and the statement also points to a superclass, which is a generalization of the subclass. For example in Figure 1 *photographic film (Q6239)* is a subclass of *data storage device (Q193395)*, *Photo equipment (Q1439598)*, and *ribbon (Q857421)*. With *subclass of (P279)* a taxonomy can be created in Wikidata. Figure 2 shows a fragment of Wikidata's taxonomy with a focus on the class *photographic film (Q6239)*. Taxonomies like this can be used for different tasks. Pekar and Staab [13] for example develop 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. According to this view, we would assume that a good taxonomy has only very few, possibly only one root class. The last remaining root class in Wikidata should be *entity (Q35120)*.

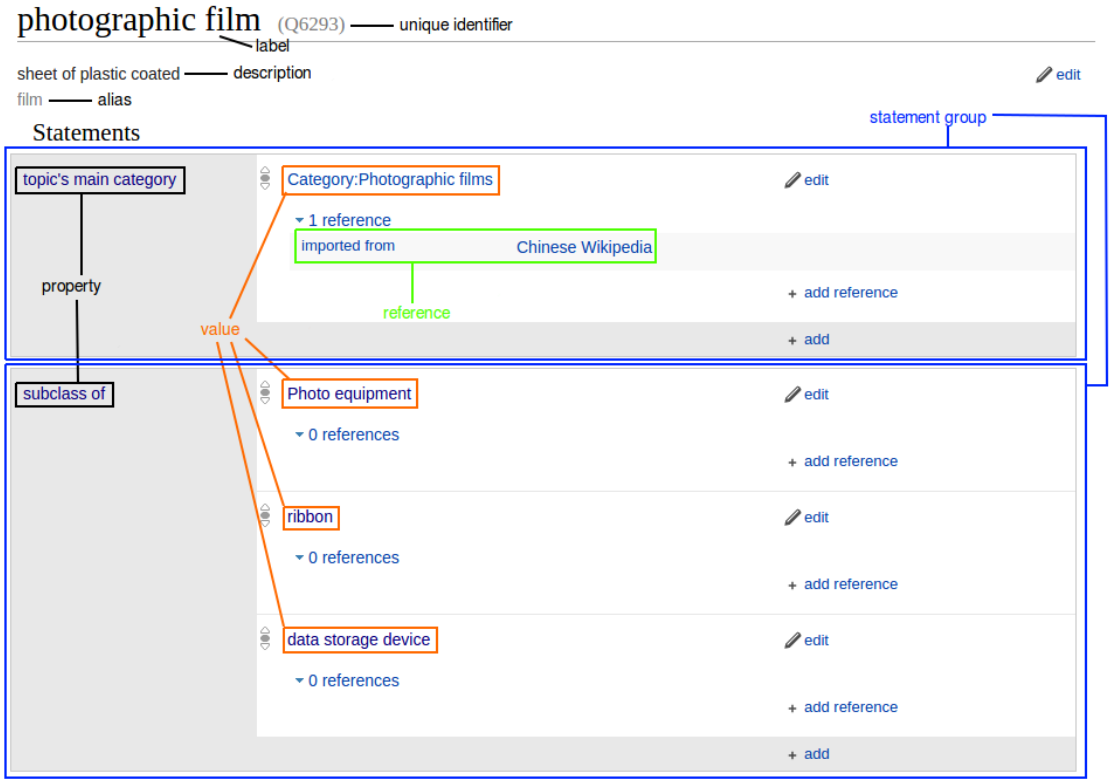

Figure 1: photographic film (Q6239)

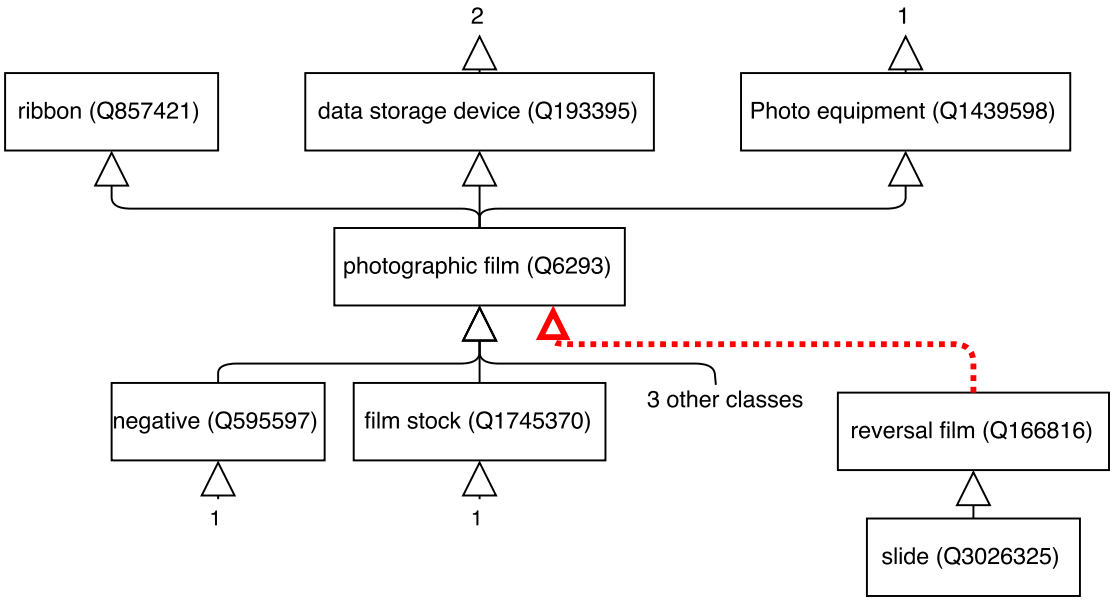

Figure 2: Fragment of Wikidata taxonomy with suggested improvement.

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. Consider for example *reversal film (Q166816)* in the taxonomy fragment of Figure 2. We can see that the class only has one subclass and is otherwise isolated in the taxonomy. Based on an expert opinion or the associated Wikipedia article, we can easily identify *photographic film (Q6239)* as a possible superclass of *reversal film (Q166816)*, as indicated by the red arrow. 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. Even though it is possible to solve this task by hand, which is the current process in Wikidata, multiple obstacles prevent this process to be efficient. First the number of root classes is high, and identifying them is not directly supported by Wikidata. Additionally finding an appropriate superclass for a given class is a difficult task, because the number of potential solutions is very high. Tools, which solve this task, may help the Wikidata community in improving the existing taxonomy. Similar tasks in the field of ontology learning are already well researched. We propose the use of neural networks for solving this task, because the application of them in ontology learning is sparse in existing work, and as shown in Section 3 neural networks seem to be appropriate for the task.

2 Problem statement

To define the problem following definitions are needed:
TODO 1: Redefine statement and class.

Definition 1 (Statement)**.** A statement is tuple $(pid, value)$:

- $pid \in \mathbb{N}$, which is a numerical Wikidata property ID;
- $value$ depends on the data type corresponding with pid, in most cases it will be natural number, representing a Wikidata item id.

Definition 2 (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 3 (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.

Because a class can have multiple superclasses, a tree structure is insufficient for modeling the taxonomy.

Definition 4 (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 5 (Root class)**.** Let $out(r)$ be the set of all outgoing edges of r. Let $T = (C, S)$ be a taxonomy.
$r \in C$ is called root class of T, if $|succ(r)| = 0$.
$root(T) = \{r \in C \mid |out(r)| = 0\}$ is the set of all root classes in T.

Definition 6. Define a function $sim : Class \times Class \mapsto (0, 1)$ as the similarity between two classes. Two classes have high similarity if the output of the function is close to 1.

Finally we can define our problem as the following task:

Problem. We simplify the taxonomy refinement task to the problem of finding the closest superclass for a given root class.

Given the input $W = (C, S)$ a taxonomy and $r \in C$ a root class in W,
find a function $f : Taxonomy \times Class \mapsto Class$, so that it produces
an output $s = f(W, r)$, which fulfills $\neg(s \lhd_{subclass} r)$ and $s = \max_{c \in C}(sim(c, r))$.

3 Related work

The related work for this thesis can be divided in two categories. First are papers, which try to solve similar tasks, and second is the topic of neural networks, which may be used to solve the defined problem.

3.1 Similar tasks

Maedche and Staab [8] define and analyze the topic of ontology learning. Additionally a tool called *OntoEdit* was developed in the process. [8] 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**.

Pekar and Staab [13] define algorithms for classification, which exploit the structure of taxonomies. Distributional and taxonomic similarity measures on nearest neighbors are 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 algorithms used by Pekar and Staab [13] may prove useful for the defined problem.

Petrucci et al. [14] describe a recurrent neural network model for ontology learning. 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.

3.2 Neural networks

Cao et al. [2] and Sperduti and Starita [18] 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.

Mikolov et al. [11] define 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. Pekar and Staab [13] use 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.

TODO 2: add [4] [16]

4 Preliminary data analysis

Wikidata does not inherently differ between entities and classes. Therefore it is necessary to define, how classes and root classes can be identified in Wikidata. In Wikidata an entity is a class, if it has instances or has subclasses or is a subclass. A root class is a class, which is not the subclass of any other class. It has to be noted that the results of this definition may not be completely accurate, because Wikidata does not enforce how the $instanceof(P31)$ and $subclassof(P279)$ are to be used. However Wikidata is constantly curated by editors and the number of misused properties should be low, therefore we can assume that the percentage of misidentified classes is also low.
The taxonomy of Wikidata, containing 1299276 classes, was analyzed. The following

statistics about root classes were acquired:

- 16148 root classes
- 13624 root classes with English label
- 11438 root classes with an English or Simple English Wikipedia article
- $\sim$ 4.8 statement groups (properties) per root class on average (see Figure 3)
- $\sim$ 4.69 instances per root class on average (see Figure 4)
- $\sim$ 0.86 subclasses per root class on average (see Figure 5)

The 5 most frequent properties in root classes are the following (see Figure 6):

- *instance of (P31)* with 8687 occurrences
- *Freebase ID (P646)* with 7221 occurrences
- *topic's main category (P910)* with 6243 occurrences
- *Commons category (P373)* with 6183 occurrences
- *image (P18)* with 2367 occurrences

It can be seen that classes in Wikidata are used by editors mainly for the purpose of grouping instances to a concept, because the average root class has $\sim$ 5 instances and $\sim$ 70% have instances. The taxonomy itself is underdeveloped, as most root classes have less than 1 subclass, and the number of root classes is high.
This means that taxonomy-based approaches [13] may not work well. But approaches from ontology mapping using semantic similarity [4] [16] could be used, because they exploit the instances of a class as a mean to find similar concepts.
Because most classes have labels and also corresponding Wikipedia articles, another possible approach could be the use of word vector models [**?**]. The Wikipedia articles would ensure that each label will at least occur once in some context.
Based on this observations it is proposed that the set of root classes is reduced to the set of labeled root classes with at least one instance, so that all observed classes fulfill basic requirements, which can be exploited by the new algorithms. Therefore the analysis needs to be repeated on this reduced set.

5 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. It should be answered, how many classes and especially root classes are available and what their characteristics

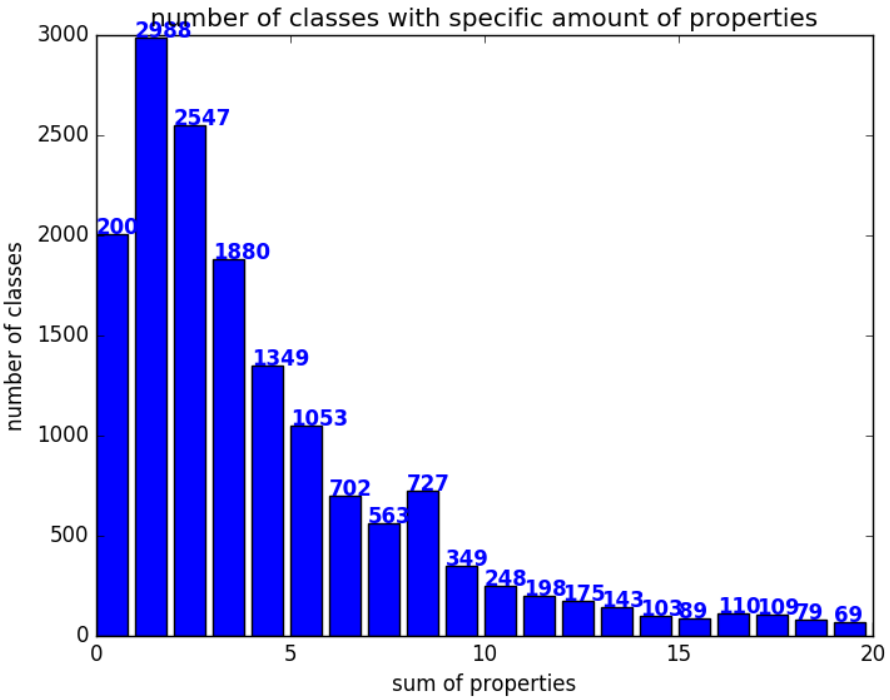

Figure 3: number of classes with a specific amount of properties

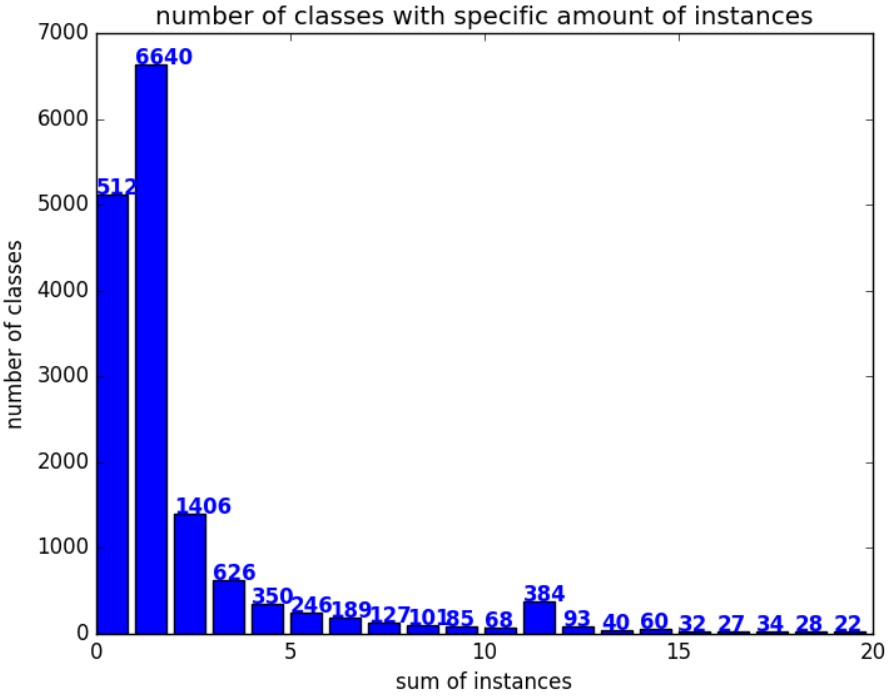

Figure 4: number of classes with a specific amount of instances

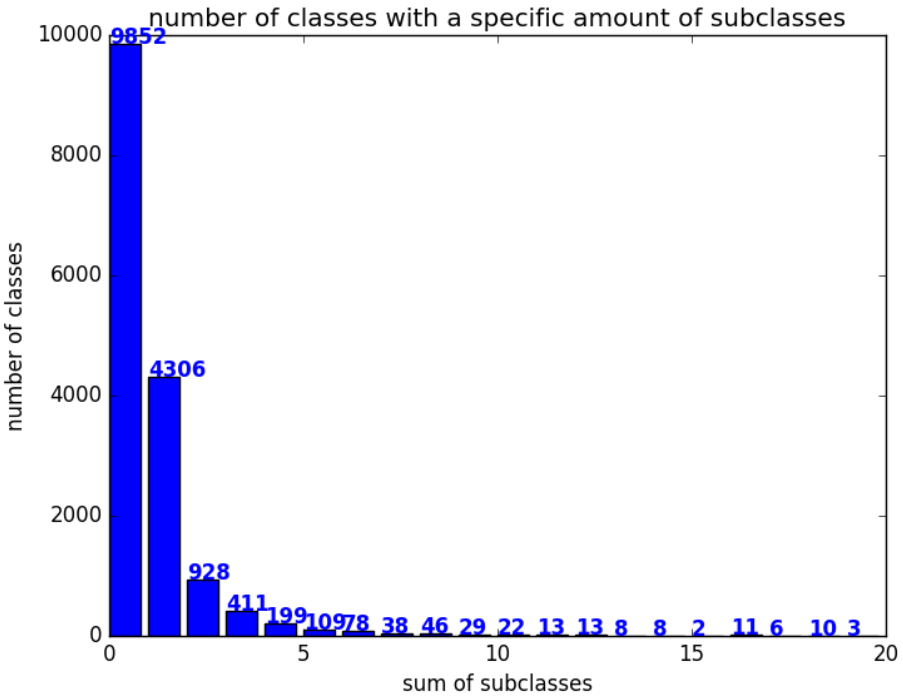

Figure 5: number of classes with a specific amount of subclasses

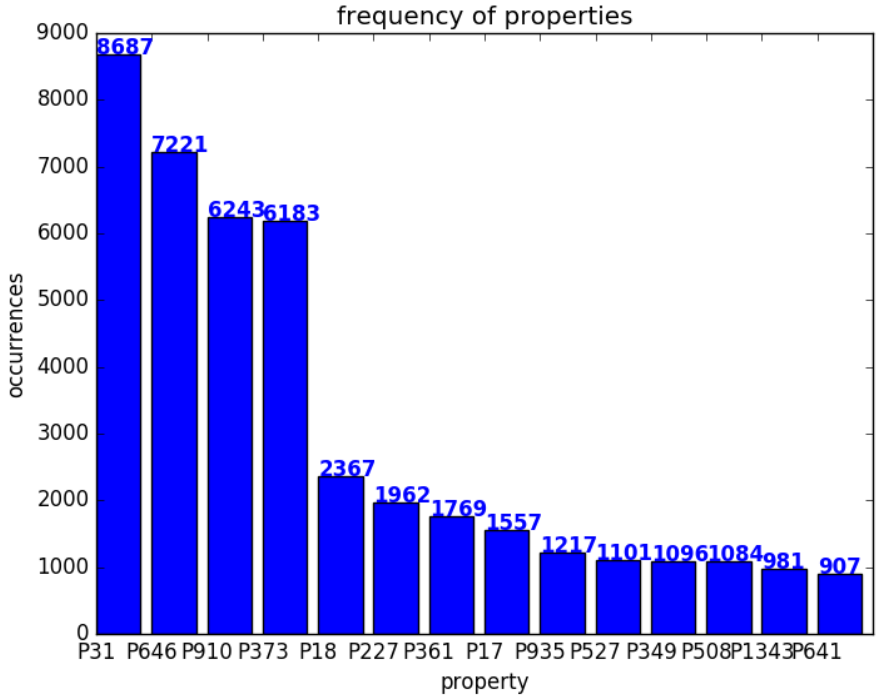

Figure 6: frequency of properties

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.

TODO 3: More specific After the neural network is developed, the system needs to be 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.

TODO 4: Compare 3 different evaluation approaches. Expert-curated ground truth, automatically generated ground truth, community agreement In the last step the develop base-line method and modified versions will be evaluated. One version will ignore the taxonomic structure, and the other methods will use different similarity measures. The evaluation will be based on a modified version of the taxonomy. A high **TODO 5: set a number** number of subclasses will be chosen, and their connection to their superclasses removed, effectively turning them into root classes. The methods will be applied on this subclasses and the correct result will be the original superclass. The methods will be compared regarding precision, recall, F_1-score, direct-hit, and near-hit ratio, where near-hit means that the result is the subclass or superclass of the correct class. This will identify the most suitable similarity measure for the given problem, and show how relevant the taxonomic structure is for the decision making. The near-hit ratio will tell if the developed methods are able to identify the correct section of the taxonomy, and if this corresponds with the consideration of the taxonomic structure.
The best performing method of the previous evaluation will be applied on all current root classes. In this evaluation participants will be asked to rate the results. The evaluation results will be analyzed and possible improvements for the network discussed.

6 Expected results

The bachelor thesis will generate the following products:

- Statistics about Wikidata's taxonomy with focus on root classes
- Neural-network based algorithm for taxonomy refinement with multiple variations

- Training and test data sets for developed algorithm
- Evaluation regarding precision, recall, F_1-score, and near-hits of base-line algorithm and variations
- Survey with Wikidata community regarding relevance of developed algorithm

At the current time I expect the algorithm to consist of two stages. In the first stage (mapping), the class will be represented as one, or multiple, vectors using neural networks. In the second stage (classification), a similarity-based classification method like k-nearest-neighbors or the, by Staab and Pekar [13], proposed algorithms can be used for finding the most similar superclass.
The first approach (see Figure 7) consists of two neural networks, which will represent the class with two vectors. The first network being the Continuous Bag-of-Words (CBOW) by Mikolov et al. [11] , and the second being graph representation neural network (GRNN), like the deep neural network by Cao et al. [2]. CBOW will be trained on the Wikipedia articles of all classes, and GRNN will be trained on the Wikidata taxonomy. To find a superclass for a given class, the closest classes in the word and in the graph vector space will be identified and used in a classification method like k-nearest neighbors.
The second approach (see Figure 8) will only use CBOW to map the classes to word vectors. But the taxonomic structure will still be exploited by using for example tree-ascending+kNN [13] in the classification stage.
For both approaches it will be necessary to calculate the representation vectors for all classes in the taxonomy ahead of time, so the suggested classification methods can be used.

If the evaluation shows that ignoring the taxonomic structure only slightly influences the quality of results, it may be a good decision to just remove the according part of the algorithm, which could greatly affect its runtime depending on the used network. I assume that the choice of similarity measure could greatly influence the evaluation results.

7 Time plan

The following outline is proposed for the thesis:

1. Foundations: definitions; problem statement; types of neural networks
2. Taxonomy analysis: statistics about classes and root classes in Wikidata
3. Related work: ontology learning; similarity measures; neural network models
4. Comparison of neural networks: compare input, output, task, type and suitability of different models for the given task

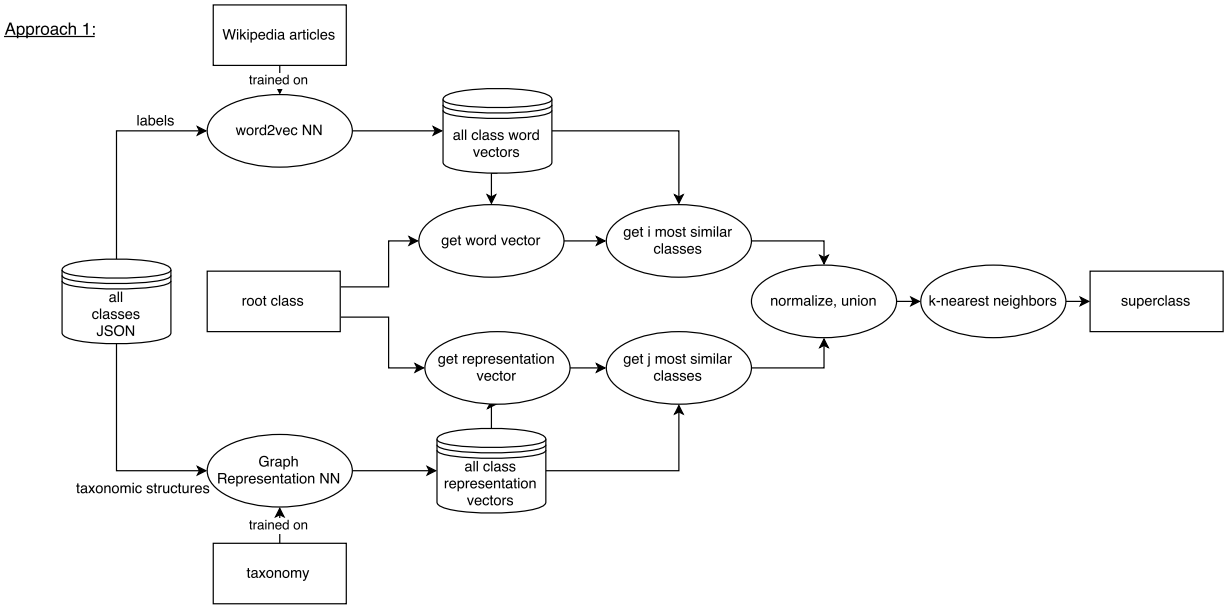

Figure 7: Solution using two neural networks [11] [2] and k-nearest neighbors for classification.

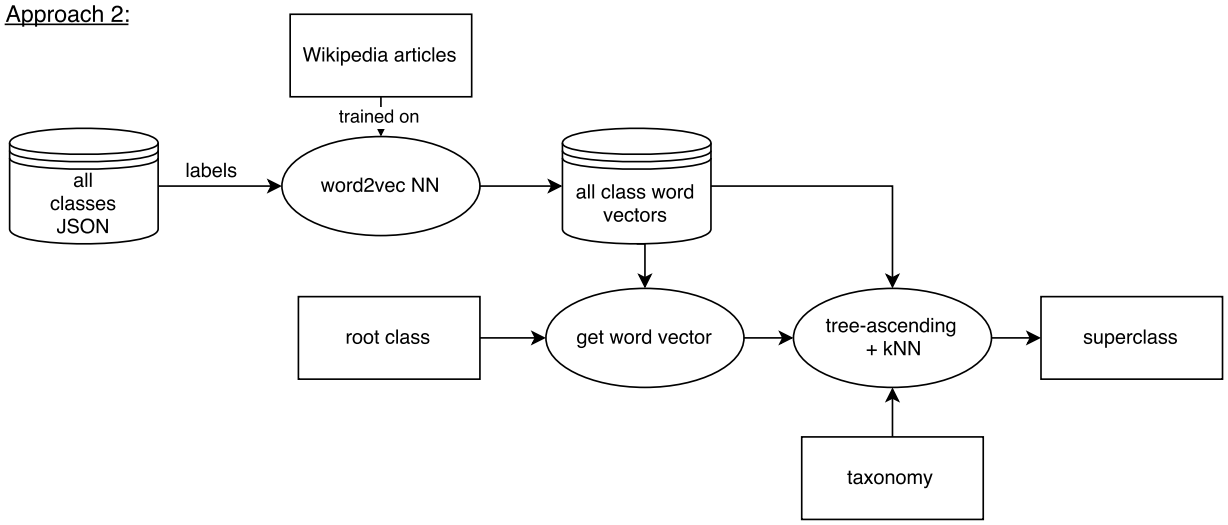

Figure 8: Solution using CBOW [11] for word representation and tree-ascending [13] for classification.

5. Development of new algorithms: design new baseline-algorithm and variations; justify design decisions based on previous sections

6. Evaluation: explain evaluation method; evaluate/compare baseline-algorithm and variations

The thesis will be written in parallel to the design and implementation of the solution. The first phase consists of research of related work and an analysis of the Wikidata taxonomy. In the second phase a baseline-algorithm and variations of it are designed and developed using the results of the previous phase. In the next phase test data is collected and the evaluation of the algorithms executed. The evaluation results will be visualized and interpreted. Finally the summary, introduction, and conclusion of the thesis will be written and a presentation prepared.

See the following Gantt chart for the time plan 9:

List of Figures

References

[1] 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. URL `http://dl.acm.org/citation.cfm?id=2390940.2390943`.

[2] 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. URL `http://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#CaoLX16`.

[3] Claudia d'Amato, Steffen Staab, Andrea G. B. Tettamanzi, Tran Duc Minh, and Fabien L. Gandon. Ontology enrichment by discovering multi-relational association

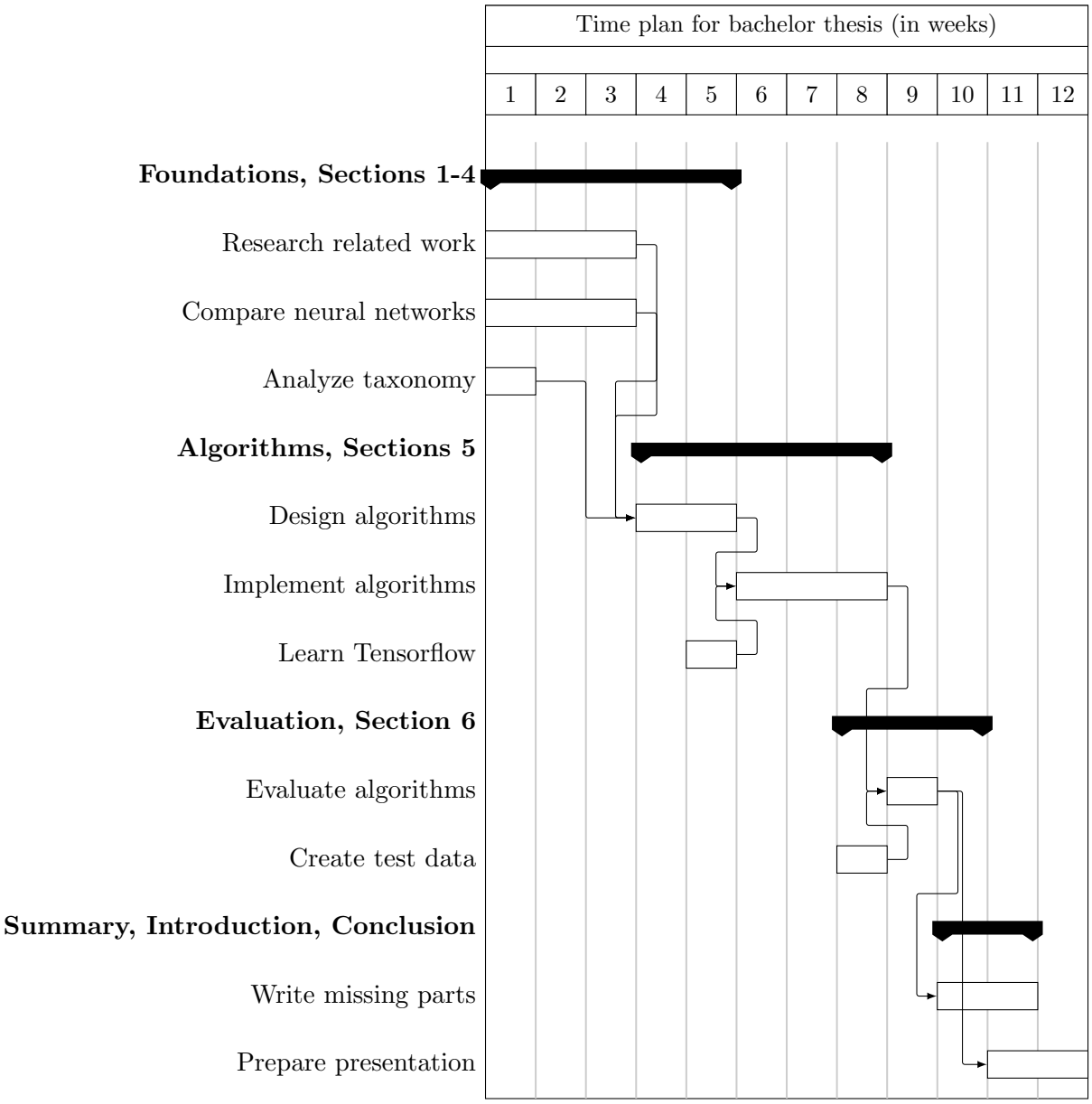

Figure 9: time plan

rules from ontological knowledge bases. In Sascha Ossowski, editor, SAC, pages 333–338. ACM, 2016. ISBN 978-1-4503-3739-7. URL http://dblp.uni-trier.de/db/conf/sac/sac2016.html#dAmatoSTMG16.

[4] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to map between ontologies on the semantic web. In Proceedings of the 11th International Conference on World Wide Web, WWW '02, pages 662–673, New York, NY, USA, 2002. ACM. ISBN 1-58113-449-5. doi: 10.1145/511446.511532. URL http://doi.acm.org/10.1145/511446.511532.

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

[6] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. CoRR, abs/1404.2188, 2014. URL http://arxiv.org/abs/1404.2188.

[7] Dekang Lin. An information-theoretic definition of similarity. In Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL http://dl.acm.org/citation.cfm?id=645527.657297.

[8] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. IEEE Intelligent Systems, 16(2):72–79, March 2001. ISSN 1541-1672. doi: 10.1109/5254.920602. URL http://dx.doi.org/10.1109/5254.920602.

[9] Tomas Mikolov. word2vec project. https://code.google.com/archive/p/word2vec/, 2013.

[10] 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. URL http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

[12] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. CoRR, abs/1605.05273, 2016. URL http://arxiv.org/abs/1605.05273.

[13] 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.

[14] Giulio Petrucci, Chiara Ghidini, and Marco Rospocher. Using recurrent neural network for learning expressive ontologies. CoRR, abs/1607.04110, 2016. URL http://arxiv.org/abs/1607.04110.

[15] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. ArXiv e-prints, June 2016.

[16] M. Andrea Rodríguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. IEEE Trans. on Knowl. and Data Eng., 15(2):442–456, February 2003. ISSN 1041-4347. doi: 10.1109/TKDE.2003.1185844. URL http://dx.doi.org/10.1109/TKDE.2003.1185844.

[17] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. IEEE Transactions on Neural Networks, 20(1):61–80, jan 2009. ISSN 1045-9227. doi: 10.1109/TNN.2008.2005605. URL http://ieeexplore.ieee.org/document/4700287/.

[18] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks, 8(3):714–735, may 1997. ISSN 10459227. doi: 10.1109/72.572108. URL http://ieeexplore.ieee.org/document/572108/.

[19] 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. ISSN 0360-0300. doi: 10.1145/2333112.2333115. URL http://doi.acm.org/10.1145/2333112.2333115.

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

[21] Min-Ling Zhang and Zhi-Hua Zhou. A k-Nearest Neighbor Based Algorithm for Multi-label Classification. volume 2, pages 718–721 Vol. 2. The IEEE Computational Intelligence Society, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1547385.