Refinement of the Wikidata taxonomy with neural networks

Proposal for Bachelor thesis

Alex Baier
abaier@uni-koblenz.de

December 6, 2016

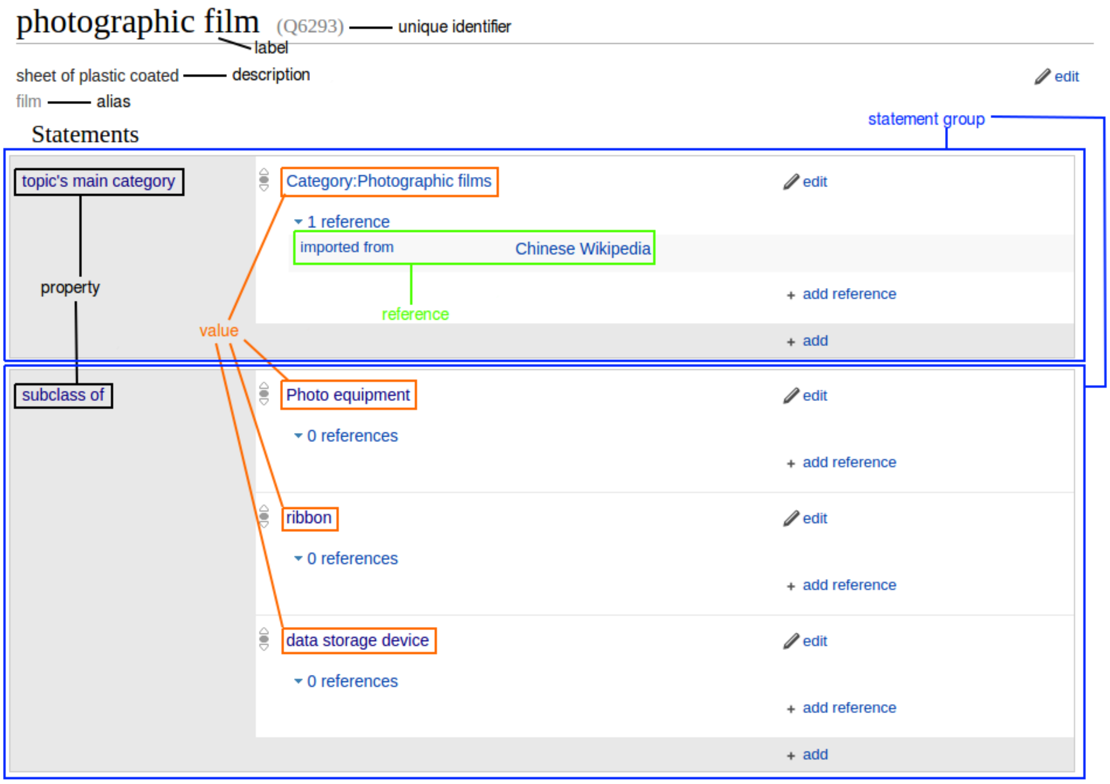
1 Motivation

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. [10] 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. 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)*.



photographic film (Q6293)

sheet of plastic coated

film

edit

Statements

topic's main category

Category:Photographic films — edit

1 reference

imported from — Chinese Wikipedia

+ add reference

+ add

subclass of

Photo equipment — edit

0 references

+ add reference

ribbon — edit

0 references

+ add reference

data storage device — edit

0 references

+ add reference

+ add

Figure 1: photographic film (Q6239)

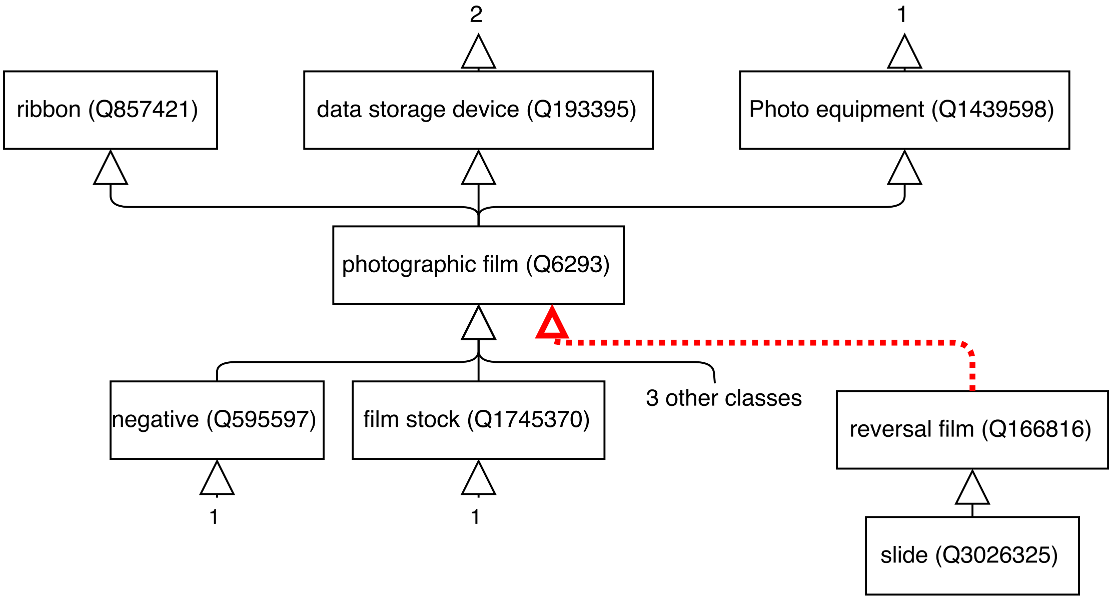


Figure 2: Fragment of Wikidata taxonomy with suggested improvement.

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 4 neural networks seem to be appropriate for the task.

2 Preliminary data analysis

The taxonomy of Wikidata, containing 1217733 classes, was analyzed. The following statistics about root classes were acquired:

- 7142 root classes
- 5332 root classes with English label
- 4590 root classes with an English or Simple English Wikipedia article
- $\sim$ 2 statement groups (properties) per root class on average (see Figure 3)
- $\sim$ 4.15 instances per root class on average (see Figure 4)
- $\sim$ 0.68 subclasses per root class on average (see Figure 5)

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

- *Freebase ID (P646)* with 3007 occurrences
- *topic's main category (P910)* with 2343 occurrences
- *Commons category (P373)* with 2012 occurrences
- *GND ID (P227)* with 744 occurrences
- *part of (P361)* with 672 occurrences

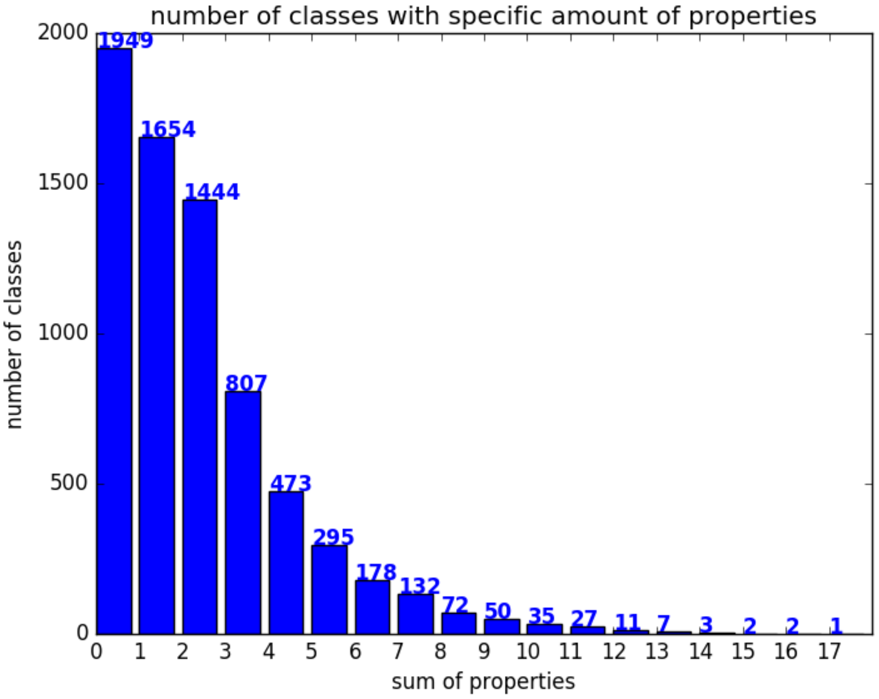


Figure 3: number of classes with a specific amount of properties

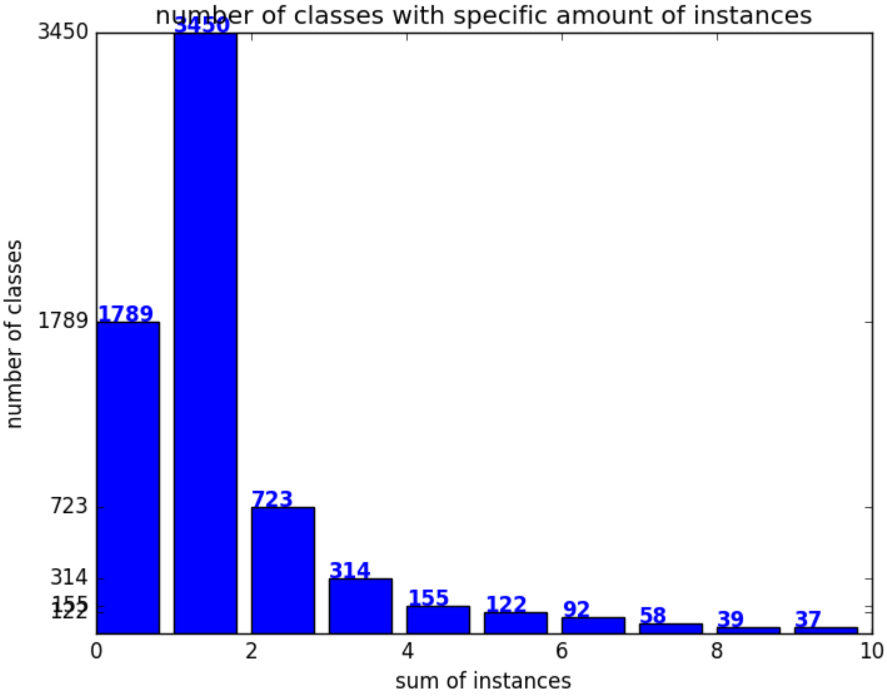


Figure 4: number of classes with a specific amount of instances

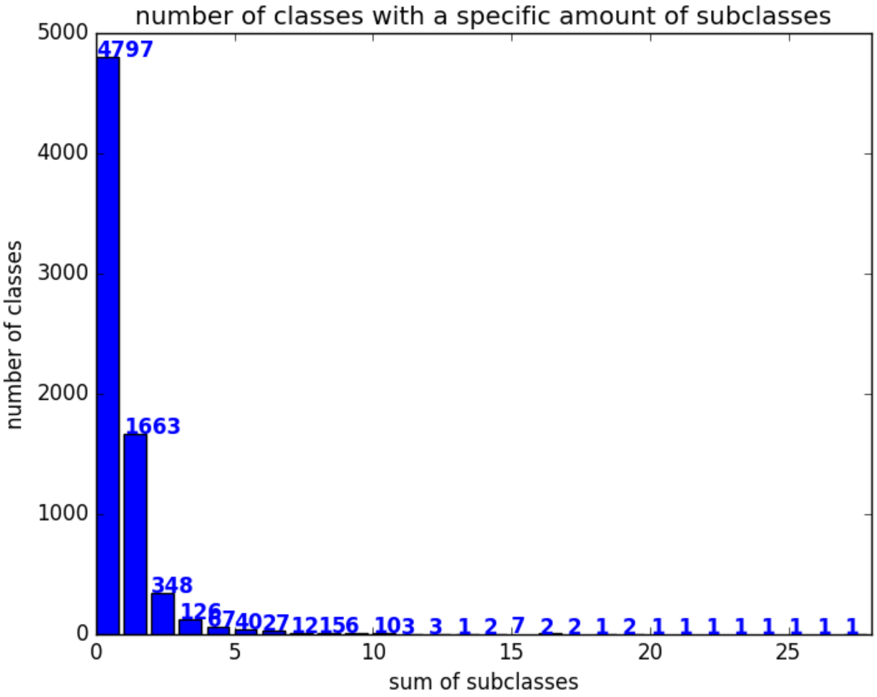


Figure 5: number of classes with a specific amount of subclasses

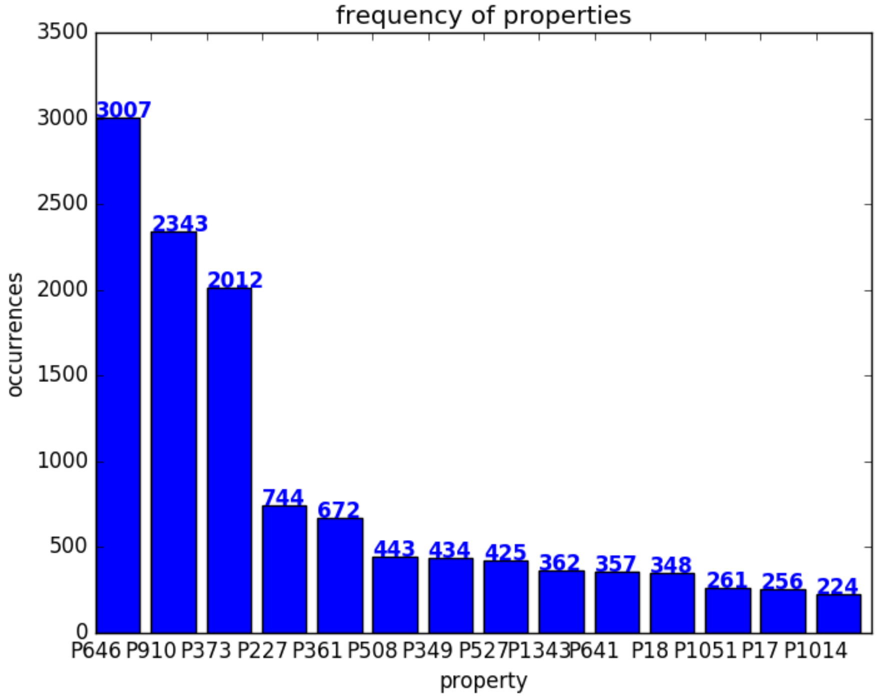


Figure 6: frequency of properties

We can see that most root classes contain very few, if any, properties, and at the same time the properties, which can be found frequently, are mostly identifiers or Wikimedia categories. This leads to the conclusion that statements won't be of much use in refining the taxonomy, and other information needs to be used. Almost 75% of the root classes are labeled and a significant amount have a corresponding Wikipedia article. Therefore we propose to only consider labeled classes for the in Section 3 defined problem. Accordingly the analysis has to be repeated for labeled root classes.

3 Problem statement

To define the problem following definitions are needed:
TODO 1: Just delete Statement, because analysis said that statements won't be of much use for solving the task.

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 $s = \max_{c \in C}(sim(c, r))$.

4 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.

4.1 Similar tasks

Maedche and Staab [7] define and analyze the topic of ontology learning. Additionally a tool called *OntoEdit* was developed in the process. [7] 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.

[10] by Pekar & Staab 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 similarity measures used in [10] may prove useful for the defined problem.

A recurrent neural network model for ontology learning is described in [11] 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.

4.2 Neural networks

Both [2] 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.

[8] 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. [10] 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.

4.3 Conclusion

For the specific task of taxonomy refinement we propose the use of neural networks, because they are able to handle structured [2] as well as textual [8] data, which is available in the Wikidata taxonomy. Additionally neural networks have shown to be achieve better results in word similarity [2] and speech recognition [9] than other methods .

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 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 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.

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 2: 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.
Additionally an evaluation with the Wikidata community will be executed. 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.

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.

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 [10], 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. [8] , 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 [10] 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.

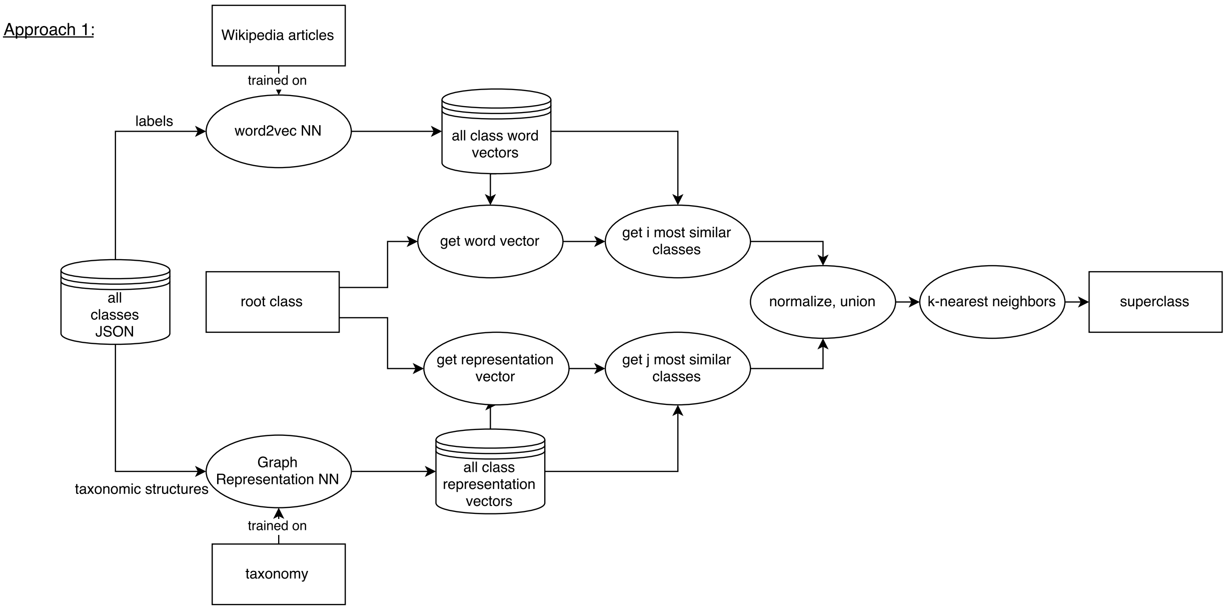


Figure 7: Solution using two neural networks [8] [2] and k-nearest neighbors for classification.

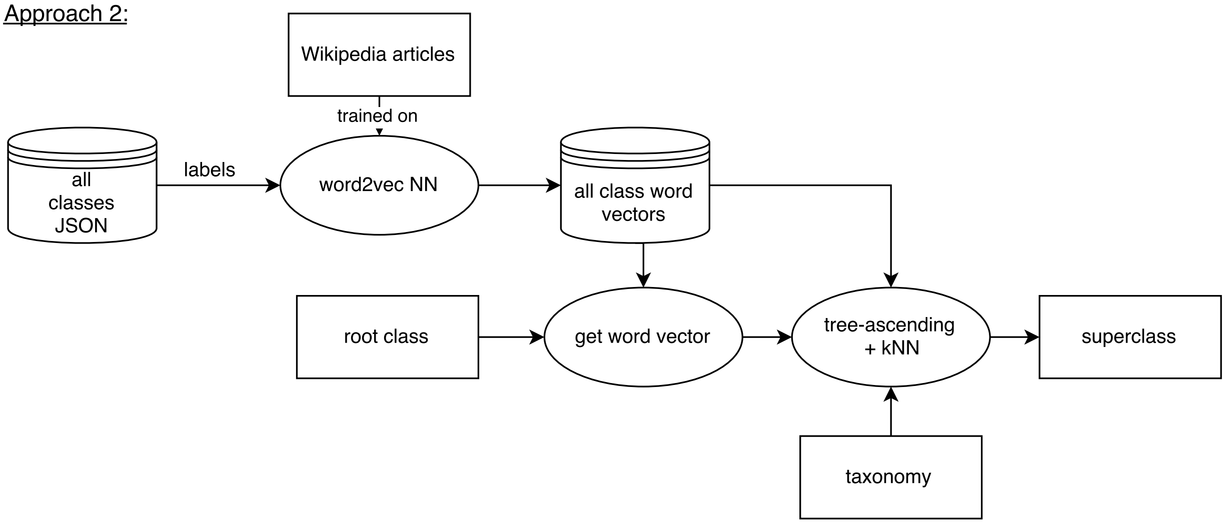


Figure 8: Solution using CBOW [8] for word representation and tree-ascending [10] for classification.

7 Time plan

Week	Tasks
1	Taxonomy analysis. Start on foundations.
2	Foundations: definitions, types of neural networks. Start related work.
3	Related work: ontology learning, similarity measures. Neural networks: Collect and prepare for next week.
4	Neural networks: analyze/ compare neural networks for task. Start development of algorithm.
5	Develop base-line algorithm, and two variations.
6	Learn how to use TensorFlow. Implement algorithms.
7	Generate training and test data. Execute tests on algorithms. Calculate precision, recall, F_1, near-hit.
8	Execute best performing algorithm on real taxonomy. Create survey for Wikidata community. Start survey.
9	Collect survey results at end of week. Write summary and introduction of thesis.
10	Interpret survey results. Write future work.
11	Create presentation. Refine thesis.
12	Final polish for presentation and thesis.

The thesis will be written in parallel to the other tasks. The tasks of the first four weeks will be highly overlapping, as they consist mainly of literature research. The implementation phase, in week 6, could lead to delays due to my inexperience with the necessary libraries like TensorFlow. The survey, in week 9 and 10, also possesses the chance of delay or even failure, as it is dependent on the participation of the Wikidata community. The last three weeks will consist of creating the presentation and refining the thesis. This phase will also act as a buffer, if it comes to the aforementioned delays.

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.

[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.

[3] 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.

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

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

[6] 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.

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[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] 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.

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