Proposal: Enrichment of ontological taxonomies using a neural network approach

Case study: Wikidata

Alex Baier
abaier@uni-koblenz.de

December 15, 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 [10] 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)*.

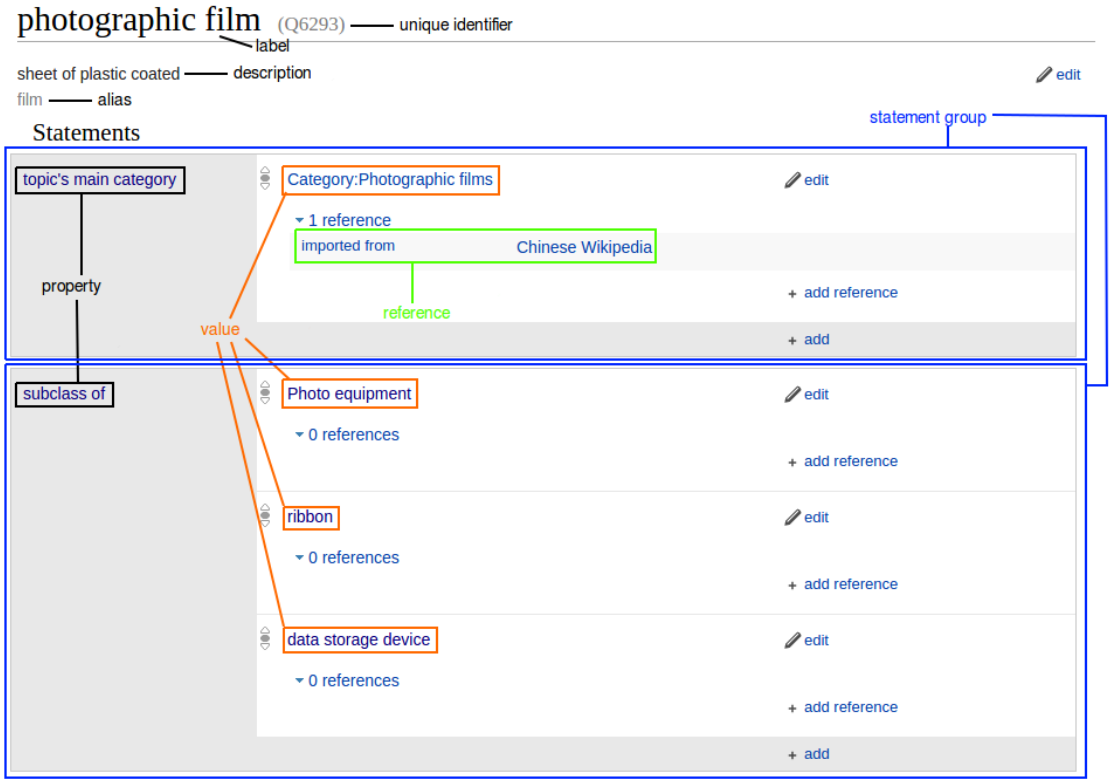

Figure 1: photographic film (Q6239)

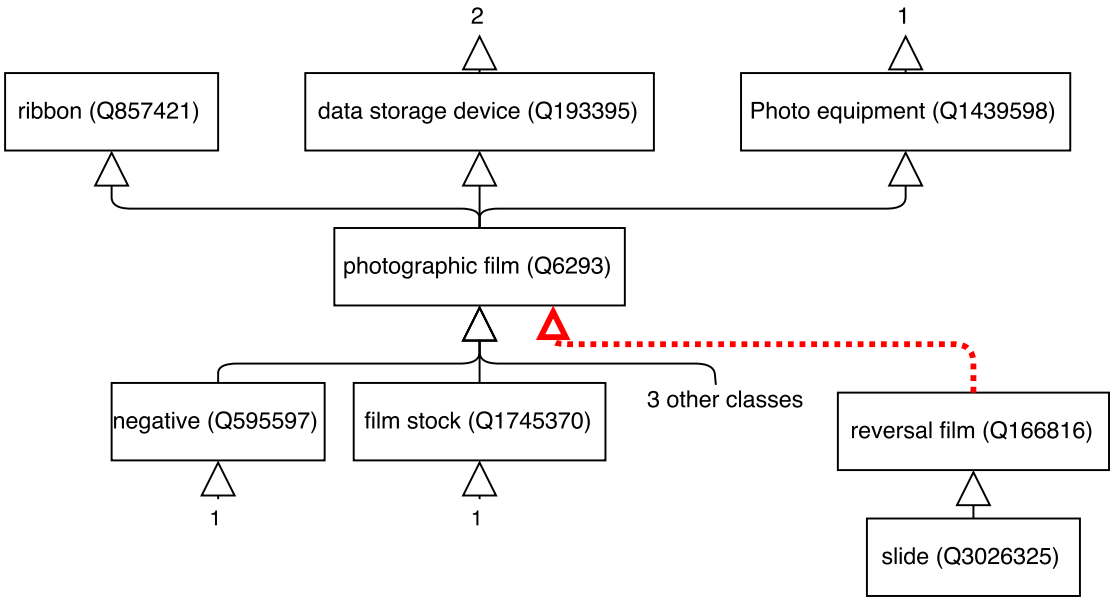

Figure 2: Fragment of Wikidata taxonomy with suggested improvement.

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:

Definition 1 (Item)**.** An item is a tuple $(id, description)$:

- $id \in \mathbb{N}$ is the numerical item ID;
- $description = (label, aliases, summary, statements, wiki)$
  - $label \in String$ is the English label of the item;
  - $aliases \in \mathcal{P}(String)$ is the set of English synonyms for the label;
  - $summary \in String$ is a short sentence describing the item;
  - $statements \in \mathcal{P}(Statement)$ is the set of statements of the item.

Definition 2 (Statement)**.** A statement is a tuple (itemid, pid, value, refs, qualifiers):

- $itemid \in \mathbb{N}$ is a numerical item ID, to which the statement belongs;
- $pid \in \mathbb{N}$ is a numerical property ID;
- $value$ is either a constant value like string, int, etc., or another item;
- $refs$ is a set of references, storing the source of information for the statement;
- $qualifiers$ is a set of qualifiers, which further specifies the statement.

If the qualifiers and references are not needed, a statement can be shortened to a triple $(itemid, pid, value)$, similar to semantic triples in RDF.
Because references and qualifiers are not used in the definition or solution of the problem, they will not be defined.

A class in Wikidata is represented as an item. However for example Doan et al. [5] model classes as a set of its instances. This is also supported by a W3C recommendation [2], which allows classes to be defined as an enumeration of instances.

Definition 3 (Class)**.** A class can be represented as the tuple $(id, description, instances)$:

- $instances \in \mathcal{P}(\mathbb{N})$ is the set of numerical item IDs, which are instances of the class.

id and $description$ are identical to the definition of Item.

Wong et al. [17] describes taxonomies as "controlled vocabulary organized into a hierarchical or parent-child structure". Typically a hierarchical structure can be defined as a tree, but because a class can have multiple superclasses, a tree structure is insufficient for modeling the taxonomy.

Definition 4 (Taxonomy)**.** A taxonomy $T = (C, S)$ is an acyclic graph, where C is a set of classes, and S is the set of edges, describing subclass-of relations between these classes.

Definition 5 (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 6 (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 7. 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. $sim(c_1, c_2) = 1$ is only the case, if c_1 and c_2 are identical.

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 [6] define and analyze the topic of ontology learning. Additionally a tool called *OntoEdit* was developed in the process. [6] 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 [10] 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 may prove useful for the defined problem, if the root classes of Wikidata are parts of the bigger components in the taxonomy graph, so that a taxonomic structure can be exploited.

Petrucci et al. [11] 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.

Doan et al. [5] describe a system for ontology mapping called GLUE, which uses machine learning techniques. Three probabilistic distribution-based similarity measures are defined. The measures are used to compare concepts of the two ontologies on a semantic level. The GLUE system takes two taxonomies as input and produces mappings for both taxonomies. Different similarity measures can be used interchangeably in the system. GLUE is evaluated with the Jacard coefficient as similarity function. The system shows to have a high matching accuracy.
The defined measures are the Jaccard coefficient, which describes an exact similarity, the "most-specific-parent" similarity, and the "most-general-child" similarity. Especially the "most-specific-parent" similarity for concepts seems to fit the thesis' problem well,

as the optimal solution for the defined problem is to the most similar, therefore most specific, superclass/parent class for a given class.

Rodríguez and Egenhofer [13] describes different matching using semantic similarity between entity classes from different ontologies. The following three measures are shown. Word matching compares the number of common and different words in the synonym sets (aliases in Wikidata) of entity classes. Feature matching uses common and different characteristics between the entities of the classes. Information content defines the similarity as degree of informativeness of the immediate superconcept that subsumes the two concepts being compared.
For the task of the thesis, word matching and feature matching are relevant, as Wikidata offers aliases for word matching, and it will be shown in Section 4 that classes in Wikidata have on average about 5 entities, which is relevant for feature matching.

3.2 Neural networks

Cao et al. [3] and Sperduti and Starita [15] 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. [8] 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. In comparison to discrete word vectors, for example used by Pekar and Staab [10], Arisoy et al. [1] state that continuous word vectors have a better notion of similarity, which means that semantically close words are also close in the continuous word vector space. Therefore the use of newer neural network language models could serve to improve older classification algorithms, like tree-ascending+kNN [10].

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 [10] may not work well. But approaches from ontology mapping using semantic similarity [5] [13] 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 [8]. 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:

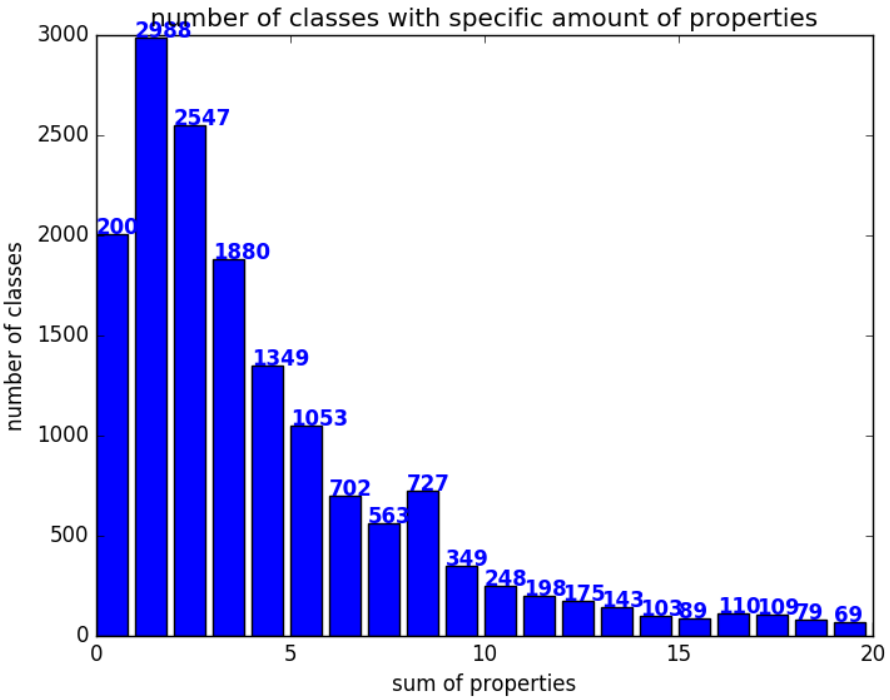

Figure 3: number of classes with a specific amount of properties

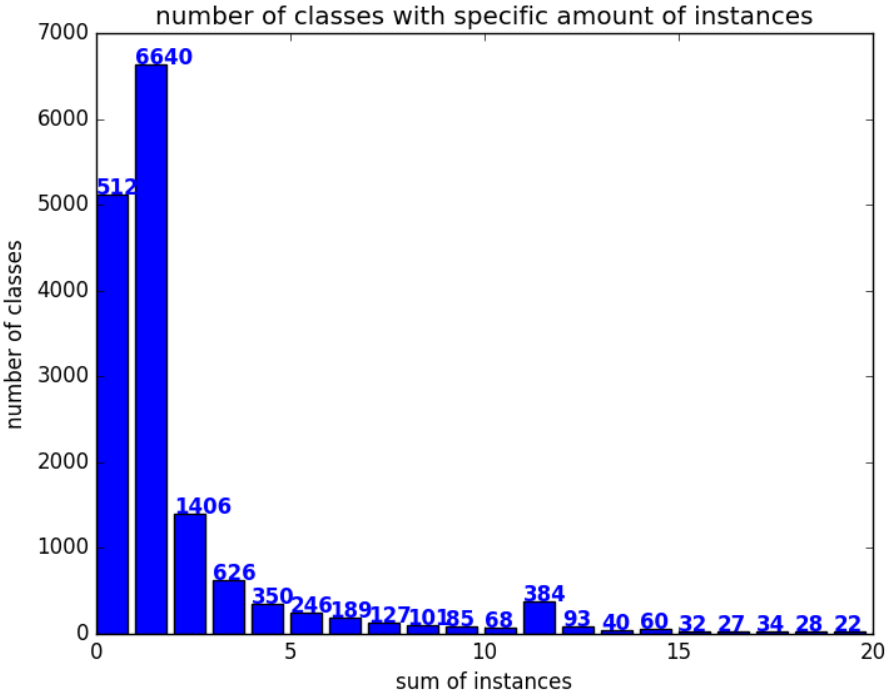

Figure 4: number of classes with a specific amount of instances

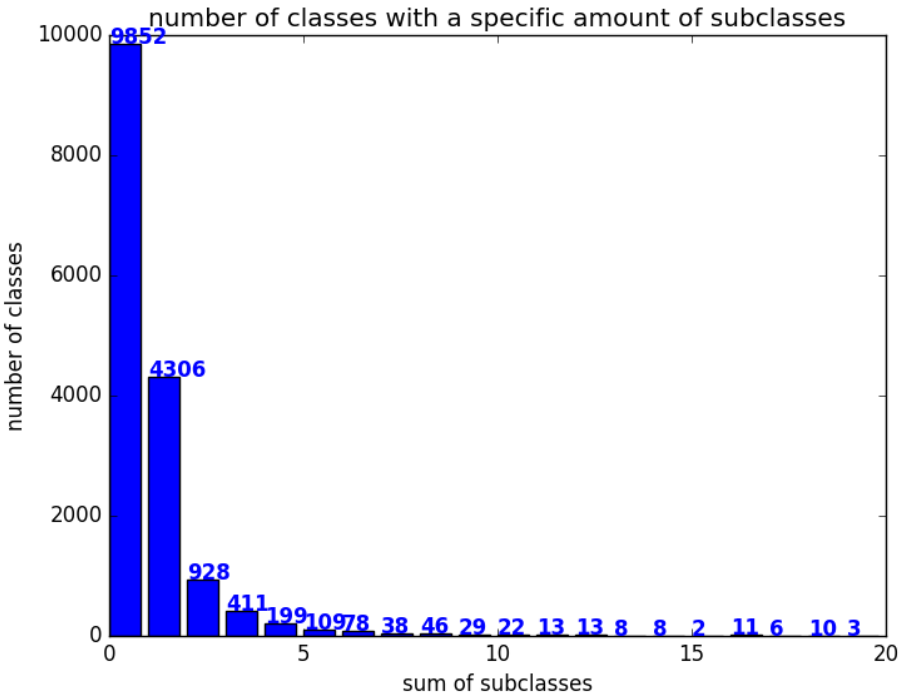

Figure 5: number of classes with a specific amount of subclasses

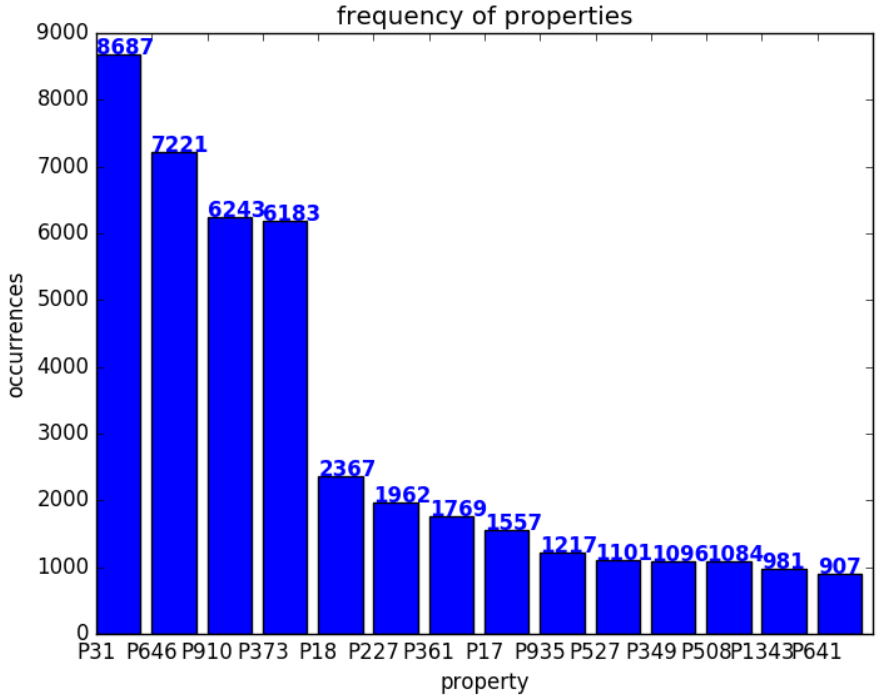

Figure 6: frequency of properties

5.1 Taxonomy analysis

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. Additionally the analysis shall be repeated on the subset of labeled root classes with instances, as proposed in Section 4. This will allow a more focused search and analysis in the following step.

5.2 Comparison of neural networks

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.

5.3 Development of a neural network-based algorithm

This leads to the development of a new algorithm using neural networks, which solves the defined problem. The algorithm should try to exploit the different aspects of the knowledge provided by Wikidata. After the baseline algorithm is developed at least one variation should also be implemented, which ignores a certain aspect of the provided knowledge, or uses a different similarity measure.

5.4 Evaluation

In the last step the baseline algorithm and its variations will be evaluated.
Variations of the baseline algorithm should only use information provided directly by Wikidata, instead of also exploiting the encyclopedic text provided by Wikidata. Other variations should use different measures of similarity for finding the most similar superclass.
There are two purposes to the evaluation. Firstly, the evaluation will show if the developed algorithm is able to solve the problem. And secondly, the evaluation of the variations will show, which measures are better suited for the task, and if it is necessary to include additional data, which is not provided by Wikidata to get good results.
Two different approaches for evaluation are proposed. For both approaches a ground truth for testing needs to be created, which consists of pairs of root classes and their correct superclasses.
In the first approach, the ground truth will be created by group of experts via for example crowdsourcing. In this case the subset of used root classes will be taken from the set of root classes in the current taxonomy.
In the second case, the ground truth will be automatically generated. It can be assumed, that the subclass relations in the current taxonomy are valid. Therefore for a set of non-root classes in the taxonomy the subclass relation to their superclasses will be removed, and the removed relation used as ground truth. The modified taxonomy will be used in

the algorithms.
For these evaluations, the percision, recall, F_1-score, direct-hit ratio, and near-hit ratio will be calculated for all versions of the algorithm.
The first approach should generate a better ground truth, and the application of the algorithms on the unmodified taxonomy is a more realistic testing scenario than the second approach using a modified taxonomy. But the first approach also requires the participation of voluntary experts, which would require a high time investment, because the experts need to be found and motivated to participate in the task.
In comparison the second approach can be executed by a single person, because the ground truth already exists. For this approach the quality of the ground truth varies depending on the number of editors for the chosen classes. The number of edits of a class correlates directly to its quality, because a high number of edits implies that many different users reached an agreement about the values of a class. Therefore it is possible to create a good ground truth with this approach by choosing only classes, which have a certain threshold of edits.
For the evaluation in the thesis, the second evaluation approach will be used, because it requires less time and should provide accurate results.

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

At the current time I expect the algorithm to use a neural network language model (NNLM). Different models based on feed-forward [8], deep [1] or recurrent [7] neural networks exist. These models map classes to continuous word vectors, which can effectively be used in similarity tasks, as stated by Arisoy et al. and Mikolov et al.. Of these models, *word2vec* by Mikolov et al., which includes CBOW and Continuous Skip-gram, seems to be a well-proven solution, as it is part of important software libraries for machine learning like TensorFlow and DeepLearning4J.
The information encoded in the classes and instances of the knowledge will be translated to text. Statements can be represented as three word sentences. The text can then be used as training data for the NNLM. Because the sentences are very short and only contain semantic information, there is almost no noise in the data and the context window of the model can be very small, which reduces the required training time.
For word vector pairs the similarity will be computed and stored in a similarity matrix,

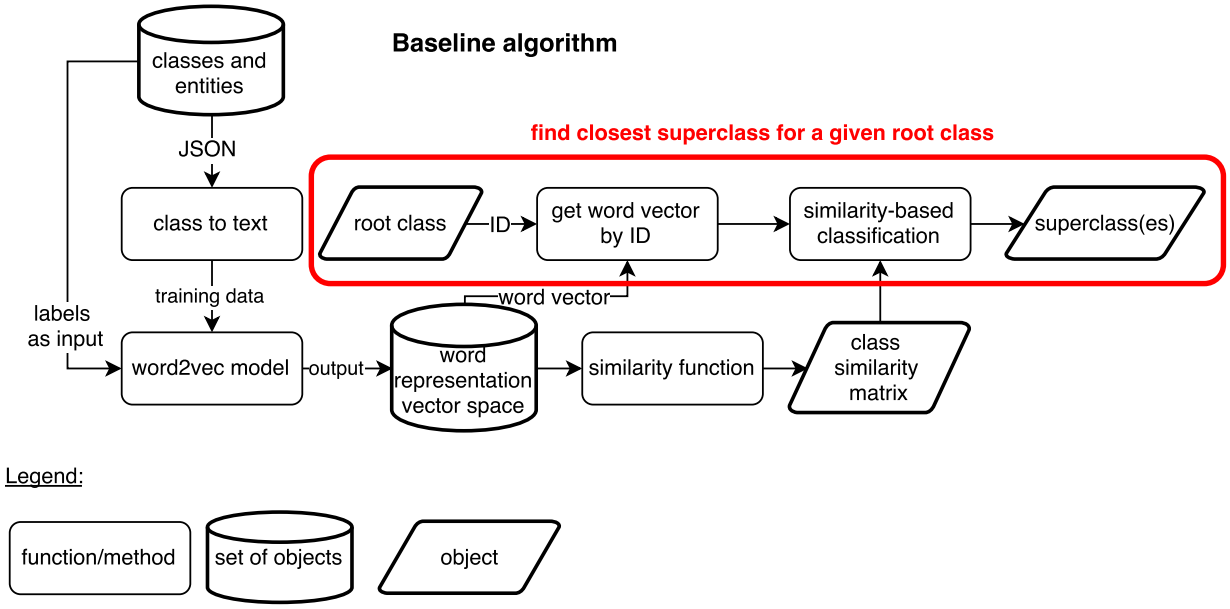

Figure 7: Baseline algorithm

this will allow faster computation in the classification. An appropriate similarity function for high-dimensional continuous vectors needs to be researched for this task.
For finding the most similar superclass, a similarity-based classification method like k-nearest-neighbors will be used.
Figure 7 is a model of the described baseline algorithm. It is only necessary to execute the methods outside of the red circle once to train the model and generate the similarity matrix and vector space. After this step, the classification can be executed repeatedly for different inputs without retraining the model.

A variation of the baseline is proposed, which uses the encyclopedic text provided by Wikipedia for training. Evaluation will show, whether the generated Wikidata data set or the bigger Wikipedia natural text data set will train the better model.
Other variations could be used to compare the performance of different similarity functions and classification methods.

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

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 Figure 8 for a Gantt chart of the time plan.

List of Figures

1 photographic film (Q6239) 2
2 Fragment of Wikidata taxonomy with suggested improvement. 2
3 number of classes with a specific amount of properties 8
4 number of classes with a specific amount of instances 8
5 number of classes with a specific amount of subclasses 9
6 frequency of properties 9
7 Baseline algorithm 12
8 time plan 14

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] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Peter F. McGuiness, Deborah L. Patel-Schneider, and Lynn Andrea Stein. Owl reference - w3c recommendation. https://www.w3.org/TR/owl-ref/, 2004.

[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

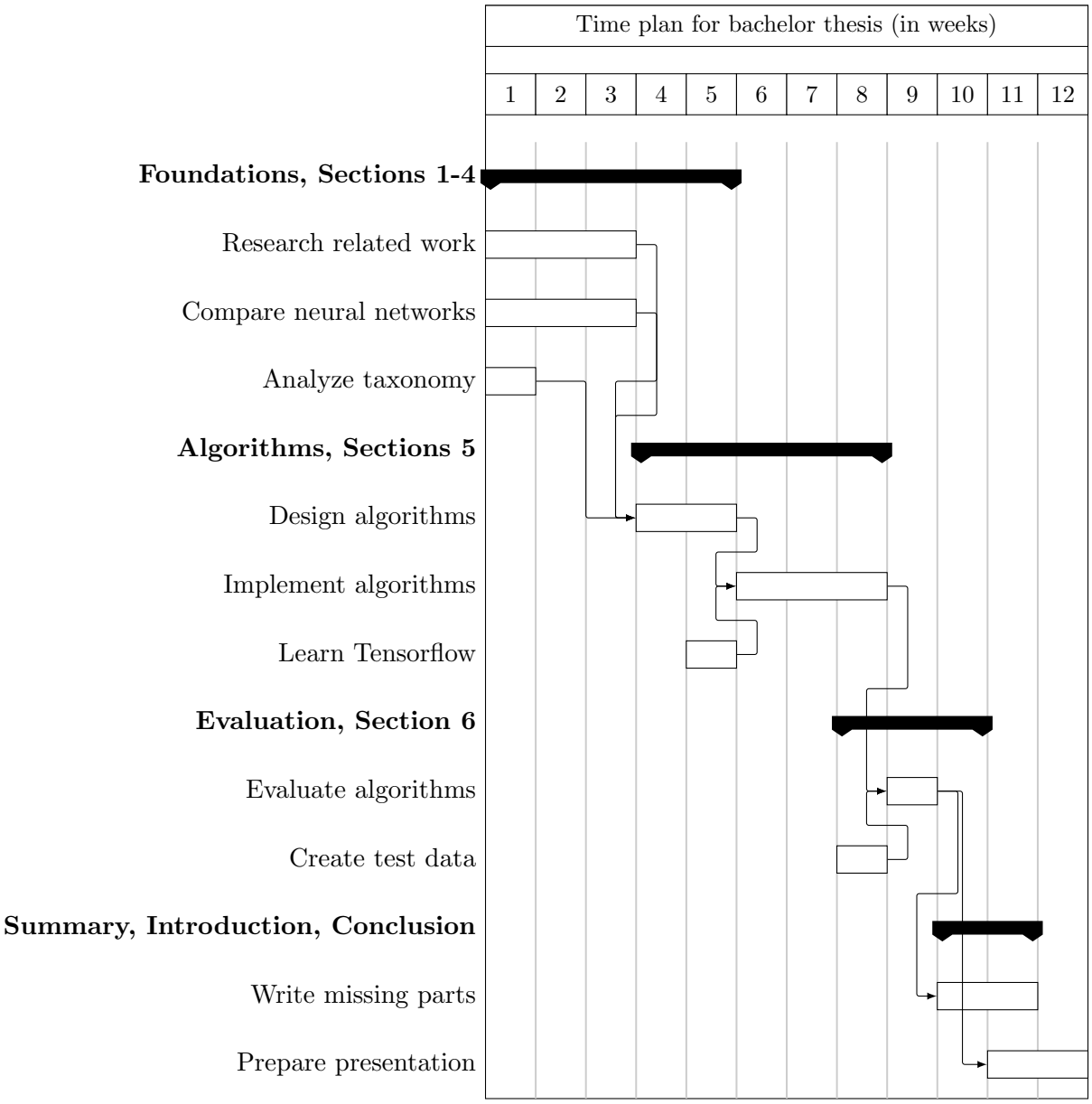

Figure 8: time plan

1145–1152. AAAI Press, 2016. URL http://dblp.uni-trier.de/db/conf/aaai/aaai2016.html#CaoLX16.

[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. ISBN 978-1-4503-3739-7. URL http://dblp.uni-trier.de/db/conf/sac/sac2016.html#dAmatoSTMG16.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[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. URL http://arxiv.org/abs/1607.04110.

[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] 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.

[14] 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/.

[15] 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/.

[16] Roger Weber. Similarity Search in High-Dimensional Vector Spaces. PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2000.

[17] 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.

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

[19] 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.