Summaries of Related work

Alex Baier
abaier@uni-koblenz.de

December 13, 2016

Contents

1 Ontology learning

1.1 Taxonomy learning – factoring the structure of a taxonomy into a semantic classification decision

[13] defines a tree-ascending algorithm, which calculates the taxonomic similarity between nearest neighbors. Evaluation shows that the algorithm is good at choosing the correct class for a new word. A combination of tree-ascending and k-nearest-neighbors was developed, which slightly improved the results in finding the correct class. Another algorithm, tree-descending, was shown to be less efficient than standard classifiers.
The paper solves a similar problem to the problem of my work. The exploitation of the taxonomy for classification proves to be effective and should therefore be incorporated into the solution of the thesis' problem.

1.2 Ontology Enrichment by Discovering Multi-Relational Association Rules from Ontological Knowledge Bases

The algorithm presented by d'Amato et al. [3] discovers relational association rules expressed SWRL, which allows the discovered rules to be directly integrated into the ontology and increase its expressiveness. This is done by detecting hidden patterns in a populated ontological knowledge base, and then obtaining relational association rules from these pattern.
Not relevant for the thesis, because the work is concerned with finding new rules in ontological KBs, while the task of the thesis is to find new connections in the taxonomy of a KB. And other papers solve problems with a higher similarity to the thesis' problem.

1.3 Ontology Learning for the Semantic Web

The task of ontology learning is analyzed by [8]. A semi-automated approach is considered, which consists of 5 steps: import/reuse existing ontologies, extract major parts of target ontology, prune to adjust ontology for primary purpose, refine ontology to complete ontology at fine granularity, and apply ontology on target applications, which validates the results. These steps may be repeated as often as necessary.
The problem of the thesis belongs to the task of ontology refinement, because an ontology is already provided.

1.4 Ontology Learning from Text: A Look Back and into the Future

[20]

1.5 Determining semantic similarity among entity classes from different ontologies

Rodríguez and Egenhofer [17] describes different matching approaches for semantic similarity between entity classes from different ontologies. Word matching compares the number of common and different words in the synonym sets of entity classes. Feature matching uses common and different characteristics between objects or entities. Information content uses information theory to define a similarity measure in terms of the degree of informativeness of the immediate superconcept that subsumes the two concepts being compared.

1.6 Learning to map between ontologies on the semantic web

Doan et al. [4] describe GLUE, a system using machine learning techniques for ontology learning. A concept is modeled as a set of instances. Three semantic similarity measures for concepts are defined, the Jaccard coefficient, the "most-specific-parent" similarity, and the "most-general-child" similarity. The system takes as input two taxonomies with data instances and returns mappings for both entered ontologies.

2 Similarity

2.1 An Information-Theoretic Definition of Similarity

[7] proposes a definition for similarity, which is independent to a particular application. Similarity is seen as a function of the commonalities and differences of two objects. The similarity measure is derived from a number assumptions about similarities. The measure is applied on ordinal values, feature vectors, words, and semantic similarity in taxonomies. All applications shows that the defined measure produces results of similar quality to domain-specific measurements.

2.2 A k-nearest neighbor based algorithm for multi-label classification

Zhang and Zhou [22] defines a lazy multi-label classification algorithm based on k-nearest neighbors.
Multi-label classification may be relevant for thesis as each class can have multiple superclasses. k-nearest neighbors algorithms use distance or similarities measures to find the k-nearest neighbors and to assign weights to each neighbor based on its distance/similarity to the input vector.

3 Neural networks

3.1 Using Recurrent Neural Network for Learning Expressive Ontologies

[15] describes a recurrent neural network with short-term memory capabilities through Gated Recursive Units for usage in ontology learning. It is argued that recurrent neural networks can handle this task, because they proved to do well in handling natural languages, and therefore should also be able to "handle the typical syntactic structures of encyclopedic text".
The ontology learning process is described as transduction task. The typical structure of encyclopedic text is exploited by translating the text into a logical formula using a pipeline, which creates a formula template and a tagged sentence. These are combined to create a formula.
This pipeline is assembled by a combination of neural networks.

- Sentence tagging: single Recurrent Neural Network with Gated Recursive Units.
  - Input: sentence in natural language, represented as n+1 symbol sequence. First n symbols are words. The last symbol is <EOS>, which indicates end of sequence. Each word has a mapping to an integer, which represents its position in the vocabulary. The input is then transformed into a sequence of context windows.
  - Output: Estimation of each tag to be the right one for k-th word. Predicted tag for k-th word is the one with the highest probability.
- Sentence transduction: Recurrent Encoder-Decoder with Gated Recursive Units.
  - Input: The same as in sentence tagging. But the context windowing is not executed.
  - Output: Set of all possible formula terms. Input and output sequences don't have the same length.
- Gated Recursive Unit provides both networks with a short-term memory effect.

3.2 Recurrent neural network based language model

[10] describes a simple recurrent network language based model and tests its performance for the task of speech recognition. It outperforms the typical n-gram model, but has a higher computational performance.
Because speech recognition has only very few similarities to the concerned data of my work, this paper has no further relevance for problem of this work. But it should be noted, that this paper shows the power of recurrent neural networks in predicting contextual data and not having a limited size of context.

3.3 Efficient Estimation of Word Representations in Vector Space

Mikolov et al. [11] propose two new neural network architectures, which compute continuous vector representations of words. It is possible to use algebraic operations on such word vectors. An example, which is mentioned in the paper $vector(''King'') - vector(''Man'') + vector(''Woman'')$ results into a vector that is very close to the representation of the word *Queen*. The paper tries to maximize the accuracy of such operations by developing their architectures to "preserve the linear regularities among words". The Continuous Bag-of-Words Model (CBOW) and the Continuous Skip-gram Model were developed. CBOW is similar to the Feedforward Neural Net Language Model. It predicts the current word based on its context. Continuous Skip-Gram in contrast predicts the words before and after the current word.
The results of different word vectors versions were compared by applying them to a similarity task. The proposed models are able to create word vectors with a higher syntactic and semantic accuracies than other existing models.
On the project page [9] of *word2vec* the authors claim that while CBOW is faster than Skip-gram, Skip-gram is better for infrequent words. Therefore it stands to reason that for the defined problem Skip-gram may be better suited than CBOW.

3.4 Deep Neural Networks for Learning Graph Representations

[2] describes a deep neural network, which creates deep graph representation model, in which a low dimensional vector is created for each vertex in the graph. The method was tested on real datasets and outperformed some state-of-the-art systems. It is able to handle noisy data, which may be important for the task of this work. The output of this network, the graph representations, can be used as input for other methods, e.g. supervised classification. Therefore this network could be used for transforming the taxonomy of Wikidata into a small-dimensional input format.

3.5 Supervised neural networks for the classification of structures

[19] describes neural networks using generalized recursive neurons for the classification of structured patterns (e.g. concept graphs). The a-priori encoding of structures for use in the neural network has drawbacks. The authors propose the use a second neural network for encoding the structure, which is trained alongside the classification network. To solve this problem the generalized recursive neuron is introduced. An encoding network is constructed using this neurons. It creates a vector, which encodes the graph, and is finally fed into a feed-forward neural network for classification. **TODO 1: Read this paper again, I think misunderstood, what the network actually does.**

3.6 Neural Networks for Classification: A Survey

[21] reviews the topic neural networks for classification. Key issues and features of neural networks are discussed. Relevant for the thesis, the topic of feature variable selection is discussed. It is important to "find the smallest set of features that can result in satisfactory predictive performance". Different approaches are discussed. It is noted that linear limitations of popular methods like principle component analysis can be overcome by using neural networks to perform networks, because they are able to perform nonlinear PCA.

3.7 A Convolutional Neural Network for Modelling Sentences

[6] defines a convolutional neural network architecture called Dynamic Convolutional Neural Network. The network models sentences using combinations of pooling and convolution layers, which internally induce something akin to parse tree on the input sentences. The specific output of the network is based on its supervised training. For example classification of sentiments was tested on movie reviews and tweets. The network proved to be effective at these tasks.

3.8 DRAW: A Recurrent Neural Network For Image Generation

Deep Recurrent Attentive Writer (DRAW) is introduced by [5]. It is an neural network architecture, which is able to generate images, by iteratively drawing and refining an image. The architecture consists of an encoder network, used for compressing images during learning, and a decoder network that recreates the image after receiving a code. This type of architecture is called variational auto-encoders. IN DRAW both of these networks are recurrent, which means a sequence of code samples is exchanged between the networks. The image generation works in multiple steps, in which different parts of the image are generated. The decision on where to write and what code samples to read is made by the neural network. After a number a step an image is produced, which cannot be distinguished from real images.

3.9 The Graph Neural Network Model

[18] develops a recursive neural network for graph classification. **TODO 2: This type of network seems to be to complex.**

3.10 On the expressive power of deep neural networks

[16] researches the the effect of depth and width of neural networks on different measures for expressivity. It is discovered that the measures, which are number of transitions,

activation patterns, and number of dichotomies, are connected exponentially to the depth, but not width, of the network. A fourth measure called trajectory length is defined, to which the other three measures are directly proportional. The trajectory length grows exponentially with the depth of the network. Additionally it is shown that the expressive power of a layer is dependent on the number of remaining layers in the network.

3.11 Deep neural network language models

Arisoy et al. [1] shows that deep neural network language models show significant improvements to single hidden layer NNLM. The notion of word similarity is missing in n-gram language models, because of their discrete nature. In comparison NNLM represent words as continuous vectors, it is estimated that therefore words that are semantically similar will also be in similar locations in the continuous vector space. The DDNLM is a feed-forward deep neural network. It is tested on textual and acoustic data, and works well. It is shown that training time is mainly based on the number of units in the network but less so on the number of hidden layers. The training takes 3 days on an 8-core CPU machine.

3.12 Learning Convolutional Neural Networks for Graphs

[12]

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 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] Yefei Peng. Ontology mapping neural network: An approach to learning and inferring correspondences among ontologies. May 2010. URL http://d-scholarship.pitt.edu/6832/.

[15] 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.

[16] 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.

[17] 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.

[18] 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/.

[19] 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/.

[20] 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.

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

[22] 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.