WebIsAGraph: A Very Large Hypernymy Graph from a Web Corpus

Stefano Faralli[1], **Irene Finocchi** [2], **Simone Paolo Ponzetto**[3], **Paola Velardi**[2]
[1]University of Rome Unitelma Sapienza, Italy
stefano.faralli@unitelmasapienza.it
[2] University of Rome Sapienza, Italy
irene.finocchi@uniroma1.it velardi@di.uniroma1.it
[3] University of Mannheim, Germany
simone@informatik.uni-mannheim.de

Abstract

In this paper, we present WebIsAGraph, a very large hypernymy graph compiled from a dataset of *is-a* relationships extracted from the CommonCrawl. We provide the resource together with a Neo4j plugin to enable efficient searching and querying over such large graph. We use WebIsAGraph to study the problem of detecting polysemous terms in a noisy terminological knowledge graph, thus quantifying the degree of polysemy of terms found in *is-a* extractions from Web text.

1 Introduction

Acquiring concept hierarchies, i.e., taxonomies from text, is a long-standing problem in Natural Language Processing (NLP). Much previous work leveraged lexico-syntactic patterns, which can be either manually defined (Hearst, 1992) or automatically learned (Shwartz et al., 2016). Pattern-based methods were shown by (Roller et al., 2018) to outperform distributional methods, and can be complemented with state-of-the-art meaning representations such as hyperbolic embeddings (Nickel and Kiela, 2017) to infer missing *is-a* relations and filter wrong extractions (Le et al., 2019). Complementary to these efforts, researchers looked at ways to scale hypernymy detection to very large, i.e., Web-scale corpora (Wu et al., 2012). Recently, (Seitner et al., 2016) applied Hearst patterns to the CommonCrawl[1] to produce the WebIsaDb. Using Web corpora makes it possible to produce hundreds of millions of *is-a* triples: the extractions, however, include many false positives and cycles (Ristoski et al., 2017).

Methods for hypernym detection like, e.g., pattern-based approaches, have a limitation in that they do not necessarily produce proper taxonomies (Camacho-Collados, 2017): automatically detected *is-a* relationships, on the other hand, can be used as input to taxonomy induction algorithms (Velardi et al., 2013; Faralli et al., 2017; Faralli et al., 2018, *inter alia*). These algorithms rely on the topology of the input graph, and, therefore, cannot be applied 'as-is' to Web-scale resources like WebIsaDb, since this resource merely consists of a set of triples. Moreover, WebIsADb does not contain fully semantified triples, i.e., subjects and objects of the *is-a* relationships consist of potentially ambiguous terminological nodes. This is because, due to their large size, source input corpora like the CommonCrawl cannot be semantified upfront. Linking to the semantic vocabulary of a reference resource like DBpedia (Hertling and Paulheim, 2017) also barely mitigate this problem, since Wikipedia-centric knowledge bases have not, and cannot be expected to have, complete coverage over Web data (Lin et al., 2012).

In this paper, we present an initial solution to these problems by building the first very large hypernymy graph, dubbed WebIsAGraph, built from *is-a* relationships extracted from a Web-scale corpus. This is a relevant task: although WordNet (and other thesauri) already provides a catalog of ambiguous terms, many nodes of WebIsAGraph are not covered in available lexicographic resources, because they are proper names, technical terms, or polysemantic words. Our graph – which we make freely available to the research community to foster further work on Web-scale knowledge acquisition – is built from the WebIsADb on top of state-of-the-art graph mining tools[2]: thanks to an accompanying plugin, it can be easily searched, queried, and explored. We-

[1]http://commoncrawl.org

[2]Neo4j: https://neo4j.com/

bIsAGraph may represent an opportunity to researchers for investigating approaches to a variety of tasks on large automatically acquired term tuples. As an example, we use our resource to investigate the problem of identifying ambiguous terminological nodes. To automatically detect whether a lexicographic node is ambiguous or not, we use information from both the graph (topological features) and textual labels (word embeddings) as features to train a model using supervised learning. Our results provide a first estimate of the degree of polysemy that can be found among *is-a* relationships from the Web.

2 Creating WebIsAGraph

We created a directed hypernymy graph from the WebIsADb (Seitner et al., 2016). WebIsADb is a Web-scale collection of noisy hypernymy relations harvested with 58 extraction patterns and consisting of 607,621,170 tuples. Since the aim of WebIsADb was to study the behaviour (on a large scale) of Hearst-like extraction patterns, rather than collecting relations with high precision, in order to reduce noise (false positives) we preselected the top-20 more precise extraction patterns in (2016) from the original 58 and identified 385,459,302 tuples.

After removing matches with a frequency lower than 3 and isolated nodes, i.e., nodes with degree equal to 0, we obtained a directed graph consisting of 33,030,457 nodes and 65,681,899 directed edges (see Table 1). The generation of such a large graph required several weeks of computation on a quad-core machine with 32 GB of RAM, using a state-of-the art graph-db system, like Neo4j. Note that the inherent sequential nature of the task of indexing tuples, nodes and edges does not benefit from the use of parallel computation. Next, we developed efficient tools for graph querying, which are released to the community, and described in `https://sites.google.com/unitelmasapienza.it/webisagraph/`, where we also include examples of queries.

3 Measuring the polysemy of WebIsAGraph

Let $p_{SI}(n)$ be the function that predicts if a terminological node n corresponds to a *monosemous* or a *polysemous* concept. We leverage a companion sense inventory as a ground truth, and we train different classifiers with a combination of topological and textual features, described hereafter.

WebIsAGraph	
nodes	33,030,457
edges	65,681,899
weakly connected components	3,099,898
nodes of largest component	26,099,001
Avg. node Degree	3.97

Table 1: Structural statistics of WebIsAGraph

Topological features. Our conjecture is that in a taxonomy-like terminological graph (even a noisy one) there is a correlation between the mutual connectivity of a node neighborhoods and its polysemy. For example, consider the polysemous word *machine* – which, according to WordNet, has at least six heterogeneous meanings, ranging from the 'any mechanical or electrical device' to 'a group that controls the activities of a political party' – and the monosemous word *floppy disk*. We expect to observe a different degree of mutual connectivity across the corresponding incoming and outgoing nodes. In particular, for monosemous words, we expect a higher mutual connectivity. With reference to Figure 1, left side, the two hypernyms of *"floppy disk"*: *"memory"* and *"data storage"*, have also *"RAM"* as a common hyponym. In contrast, nodes in the direct neighborhood of *"machine"* (leftmost graph in Figure 1) do not have mutual connections.

Our aim is thus to identify topological features that may help quantifying the previously described connectivity properties. To cope with scalability, we consider topological features built on top of 1-hop/2-hop sub-graphs of a node n. Hence, we identify two induced sub-graphs $G^{-+}(n)$ and $G^{+-}(n)$, induced on $V^{-+}(n) = In(n) \cup_{v \in In(n)} Out(v)$ and $V^{+-}(n) = Out(n) \cup_{v \in Out(n)} In(v)$ respectively, where $In(x)$ and $Out(x)$ are the sets of incoming and outgoing nodes of x (including x). Next, we remove from these sub-graphs the node n, and compute the following features:

- $cc_{G^{-+}}(n)$ and $cc_{G^{+-}}(n)$: the resulting number of weakly connected components;
- $v_{G^{-+}}(n)$ and $v_{G^{+-}}(n)$: the resulting number of nodes;
- $e_{G^{-+}}(n)$ and $e_{G^{+-}}(n)$: the resulting number of edges.

With reference to the example of Figure 1, the light gray sub-graph (a) is $G^{-+}(n)$, the dark sub-

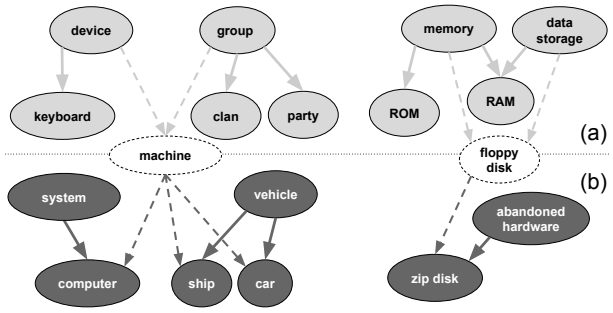

Figure 1: An example excerpt of the neighborhood induced sub-graphs for *"machine"* and *"floppy disk"*, (a) $G^{-+}(n)$ in gray and (b) $G^{+-}(n)$ in dark gray. Dashed edges connect each n with its hypernyms and hyponyms.

graph (b) is $G^{+-}(n)$, and furthermore for n = *"machine"*: $cc_{G^{-+}}(n) = 2$, $cc_{G^{+-}}(n) = 2$, $v_{G^{-+}}(n) = 5$, $v_{G^{+-}}(n) = 5$, $e_{G^{-+}}(n) = 3$, and $e_{G^{+-}}(n) = 3$, while for the n =*"floppy disk"*: $cc_{G^{-+}}(n) = 1$, $cc_{G^{+-}}(n) = 1$, $v_{G^{-+}}(n) = 4$, $v_{G^{+-}}(n) = 2$, $e_{G^{-+}}(n) = 3$, and $e_{G^{+-}}(n) = 1$.

Textual features. Similarly to topological features, our hypothesis is that textual features of the neighborhood nodes should exhibit a lower average similarity when n is polysemous. We extract textual features on top of pre-trained word embeddings, widely adopted in many NLP-related tasks (Camacho-Collados and Pilehvar, 2018). Formally, given a node n:

- $\overrightarrow{W}(n)$ is the word embedding vector of n computed as follows:

$$\overrightarrow{W}(n) = \frac{\sum_{t \in tokens(n)} \overrightarrow{we}(t)}{|tokens(n)|} \quad (1)$$

  where $tokens(n)$ is the function that retrieves the set of tokens composing the word n (e.g., if n = *hot dog*, $tokens(n) = \{$*hot* , *dog*$\}$), and $\overrightarrow{we}(t)$ is a pre-trained word embedding vector;

- $\Delta_{in}(n)$ and $\Delta_{out}(n)$: the cosine similarity between $\overrightarrow{W}(n)$ and the average word embeddings vector of incoming and outgoing nodes of n respectively;

$$\Delta_{in}(n) = CosSim(\overrightarrow{W}(n), \frac{\sum_{m \in In(n)} \overrightarrow{W}(m)}{|In(n)|}) \quad (2)$$

$$\Delta_{out}(n) = CosSim(\overrightarrow{W}(n), \frac{\sum_{m \in Out(n)} \overrightarrow{W}(m)}{|Out(n)|}) \quad (3)$$

		Features								
		topological			textual			all		
	Algo.	P	R	F_1	P	R	F_1	P	R	F_1
WordNet	Rnd	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47
		±.05	±.05	±.05	±.05	±.05	±.05	±.05	±.05	±.05
	NN	0.61	0.61	0.60	**0.72**	**0.72**	**0.72**	**0.73**	**0.73**	**0.73**
		±.02	±.02	±.02	±.03	±.03	±.02	±.04	±.04	±.04
	ABC	0.62	0.62	0.62	0.67	0.67	0.67	0.70	0.70	0.70
		±.03	±.03	±.03	±.02	±.02	±.01	±.03	±.03	±.03
	GBC	**0.62**	**0.62**	**0.62**	0.69	0.68	0.68	0.72	0.71	0.71
		±.02	±.02	±.02	±.01	±.01	±.01	±.03	±.03	±.03
DBpedia	Rnd	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
		±.03	±.03	±.03	±.03	±.03	±.03	±.03	±.03	±.03
	NN	0.60	0.60	0.59	**0.73**	**0.73**	**0.73**	**0.74**	**0.74**	**0.74**
		±.01	.01	±.01	±.03	±.03	±.03	±.03	±.03	±.03
	ABC	0.60	0.60	0.60	0.69	0.69	0.69	0.71	0.71	0.71
		±.02	±.02	±.02	±.02	±.02	±.02	±.04	±.04	±.04
	GBC	**0.61**	**0.61**	**0.61**	0.70	0.70	0.70	0.73	0.73	0.73
		±.02	±.02	±.02	±.03	±.03	±.03	±.02	±.02	±.02
WordNet∪DBpedia	Rnd	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
		±.01	±.01	±.01	±.01	±.01	±.01	±.01	±.01	±.01
	NN	0.54	0.53	0.50	**0.70**	**0.70**	**0.70**	**0.71**	**0.70**	**0.70**
		±.03	±.05	±.12	±.02	±.02	±.02	±.02	±.01	±.01
	ABC	0.55	0.55	0.55	0.64	0.64	0.64	0.65	0.65	0.65
		±0.02	±0.02	±0.02	±.01	±.01	±.01	±.02	±.02	±.02
	GBC	**0.56**	**0.56**	**0.55**	0.66	0.66	0.65	0.67	0.67	0.67
		±0.02	±0.01	±0.01	±.02	±.02	±.02	±.02	±.02	±.02

Table 2: Performance of different algorithms to detect node ambiguity.

- $Gini(n)$: sparsity index (David, 1968) of $\overrightarrow{W}(n)$.

3.1 Evaluation

Computing features. Topological features are efficiently extracted using the query tool mentioned in Section 2. To compute *textual features* (see Section 3) we use the *Glove* pre-trained word embedding vector (Pennington et al., 2014) of length 300 from the CommonCrawl.[3]

By combining these two types of features (topological and textual) we obtained three different vector input representations consisting of 6 (only topological features), 303 (only textual features) and 309 (textual and topological) dimensions respectively.

Finally, we created three "ground truth" sets of nodes in the graph for which $p_{SI}(n)$ is known. We selected a balanced number of monosemous and polysemous nouns, using the following sense inventories: i) WordNet (14,659 examples); ii) DBpedia (17,041 examples); iii) WordNet and DBpedia (31,701 examples).

Algorithms. We compared four algorithms:

- Random (*Rnd*): a random baseline which randomly classifies the ambiguity of a node;

- Neural Network (*NN*): a neural network with Softmax activation function in the output layer and dropout (Srivastava et al., 2014);

[3] https://nlp.stanford.edu/projects/glove/.

		WordNet			DBpedia			WordNet ∪ DBpedia		
	Features	$dCor$	ρ	PI weight ± std	$dCor$	ρ	PI weight ± std	$dCor$	ρ	PI weight ± std
topological	cc_{G-+}	0.593	0.185	0.0039±0.0001	**0.614**	**0.228**	**0.0628±0.0051**	0.513	**0.027**	0.0038±0.0008
	v_{G-+}	0.602	0.203	0.0022±0.0003	0.597	0.194	0.0045±0.0010	0.513	0.025	0.0025±0.0003
	e_{G-+}	0.597	0.194	0.0100±0.0016	0.600	0.200	0.0048±0.0008	**0.514**	**0.027**	0.0024±0.0001
	cc_{G+-}	0.606	0.212	0.0131±0.0013	0.579	0.159	0.0092±0.0016	0.492	-0.014	0.0049±0.0003
	v_{G+-}	**0.623**	**0.247**	0.0383±0.0035	0.580	0.159	0.0029±0.0009	0.495	-0.010	0.0013±0.0008
	e_{G+-}	0.619	0.237	0.0074±0.0010	0.583	0.167	0.0034±0.0013	0.497	-0.006	0.0054±0.0004
textual	Δ_{in}	0.379	-0.242	**0.0699±0.0036**	0.399	-0.202	0.0231±0.0023	0.433	-0.134	**0.0470±0.0027**
	Δ_{out}	0.400	-0.199	0.0101±0.0004	0.415	-0.170	0.0037±0.0015	0.431	-0.138	0.0120±0.0007
	$Gini$	0.443	-0.114	0.0042±0.0004	0.460	-0.080	0.0035±0.0009	0.494	-0.013	0.0059±0.0006
	$\overrightarrow{W}$(300 dimensions)		Avg	0.0029±0.0004		Avg.	0.0030±0.0005		Avg	0.0028±0.0003
			Min	0.0016±0.0003		Min	0.0005±0.0005		Min	0.0014±0.0003
			Max	0.0077±0.0009		Max	0.0180±0.0013		Max	0.0123±0.0011

Table 3: Distance correlation $dCor$ and Pearson coefficient ρ between polysemy and features and Permutation Importance (PI) weights (NN estimator).

- Two ensemble-based learning algorithms, namely AdaBoost (*ABC*) (Zhu et al., 2009) and Gradient Boosting (*GBC*) (Friedman, 2001): both have been shown to have high predictive accuracy (Kotsiantis et al., 2006) and are good competitors of neural methods, especially with very large datasets.

Parameter selection. Based on the Area Under Curve ROC (AUC) analysis (Kim et al., 2017), *NN* parameters have been empirically set as follows: i) when testing only with topological features (6 dimensions), we use 2 hidden layers with 4 and 2 neurons respectively and a dropout of 0.2 and 0.15; ii) when using only textual (303 dimensions), or combined textual and topological features (309 dimensions), we use 4 hidden layers, with 128, 64, 32 and 8 neurons respectively and a dropout of 0.3,0.25,0.2 and 0.15.

Results. We show in Table 2 the resulting precision, recall and F_1 of the five systems across the ground truths datasets and for the combinations of features (see Section 3). The metrics are averaged on five classification experiments, with a random split (85% train, 10% validation and 5% test) of the ground truth sets. As shown in Table 2, *NN* outperforms the others ensemble methods, obtaining a F_1 score around 0.70. The comparison of performances across the three combinations of features reveals that topological features are not enough to build a model for polysemy classification but can slightly boost the overall already compelling performances of word embeddings-based features.

In Table 3 we show the Person coefficient ρ and the distance correlation $dCor$[4], with the aim of analyzing how each feature correlates with the polysemy observed in the three ground truth dictionaries. We observed that the features with the highest correlation with polysemy are e_{G+-}, cc_{G-+} and v_{G-+} (see Section 3). Additionally we report the resulting weights of *Permutation Importance* (PI) applied to the *NN* system with the aim of measuring how the performance decreases when a feature is perturbed, by shuffling its values across training examples (Breiman, 2001). We observed that the features which most influenced the performances are $\Delta_{in}(n)$ (WordNet and WordNet∪DBpedia) and cc_{G-+} (DBpedia). Furthermore, we found that although topological features affect the performance only by a 1% in the average, a number of topologically related features, such as cc_{G-+}, v_{G-+} and e_{G+-} are shown to be indeed related with polysemy. In our future work, we plan to create an ad-hoc ground-truth sense dictionary, since especially WordNet includes extremely fine-grained senses that do not help validating our conjecture about reduced mutual connectivity and contextual similarity of a node's neighborhood in case of monosemy.

4 Conclusion

The main contribution of this work is a new resource obtained by converting a large dataset of *is-a* (hypernymy) relations automatically extracted from the Web (such as WebIsADb) into a graph structure. This graph, along with its accompanying search tools, enables descriptive and predictive analytics of emerging properties of termino-

[4] ρ and $dCor$ are indexes to estimate how two distributions are independent.

logical nodes. We used here our new resource to investigate whether a node *polysemy* can be predicted from its topological features (i.e., connectivity patterns) and textual features (meaning representations from word embeddings). The results of this preliminary study have shown that textual features are good predictors of polysemy, while topological features appear to be weaker predictors even if they have a significant correlation with the polysemy of the related node.

References

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32, Oct.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Intell. Res.*, 63:743–788.

Jose Camacho-Collados. 2017. Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations.

H. A. David. 1968. Gini's mean difference rediscovered. *Biometrika*, 55(3):573–575.

Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. 2017. The contrastmedium algorithm: Taxonomy induction from noisy knowledge graphs with just a few links. In *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 590–600. Association for Computational Linguistics.

Stefano Faralli, Irene Finocchi, Simone Paolo Ponzetto, and Paola Velardi. 2018. Efficient pruning of large knowledge graphs. In *Proc. of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4055–4063.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pages 539–545.

Sven Hertling and Heiko Paulheim. 2017. Webisalod: Providing hypernymy relations extracted from the web as linked open data. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proc., Part II*, pages 111–119.

Chulwoo Kim, Sung-Hyuk Cha, Yoo An, and Ned Wilson. 2017. On roc curve analysis of artificial neural network classifiers. In *Florida Artificial Intelligence Research Society Conference*.

S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. 2006. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, Nov.

Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings.

Thomas Lin, Mausam, and Oren Etzioni. 2012. Entity linking at web scale. In *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88. Association for Computational Linguistics.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6341–6350. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Petar Ristoski, Stefano Faralli, Simone Paolo Ponzetto, and Heiko Paulheim. 2017. Large-scale taxonomy induction using entity and word embeddings. In *Proc. of the International Conference on Web Intelligence*, WI '17, pages 81–87, New York, NY, USA. ACM.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proc. of the 56th ACL (Volume 2: Short Papers)*, pages 358–363. Association for Computational Linguistics.

Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A large database of hypernymy relations extracted from the web. In *Proc. of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proc. of the 54th ACL (Volume 1: Long Papers)*, pages 2389–2398. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3).

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proc. of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 481–492, New York, NY, USA. ACM.

Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. 2009. Multi-class adaboost. *Statistics and Its Interface*, 2(3):349–360.