Author(s): Zafar, Bushra; Imran, Ayesha; Asghar, Muhammad; Cochez, Michael; Hämäläinen, Timo

Title: Automatic Taxonomy Induction based on Word-embedding of Neural Nets

Year: 2018

Version: Published version

Automatic Taxonomy Induction based on Word-embedding of Neural Nets

[1]Bushra Zafar, [1]Ayesha Imran, [1]Muhammad Zeeshan Asghar, [1,2]Michael Cochez, [1]Timo Hämäläinen
[1]Department *of Mathematical Information Technology, University of Jyväskylä, Finland, bushra.zafar025@gmail.com, ayesha.a.imran@student.jyu.fi, muhammad.z.asghar@jyu.fi, timo.t.hamalainen@jyu.fi*
[2]F*raunhofer – FIT Sankt Augustin, Informatik 5 RWTH Aachen University DE-52056 Aachen, Germany, michael.cochez@fit.fraunhofer.de*

Abstract

Taxonomy is a knowledge management tool that presents useful information in a well-ordered structure prevents overloading of information on its access and making the information access qualitative. This article is concerned with automatically extracting asymmetrical hierarchical relations from a large corpus and subsequent taxonomy construction by domain independent and semi-supervised system. The methodology relies on the term's distributional semantics. The algorithm utilizes the word-embedding generated from the vector space model. The model is trained over a large corpus to generate word-embedding of each word in a corpus. Then, the system finds and extracts the hypernyms by using the genetic algorithm based on distributional semantics calculations. In the last step, the system adds hyponym-hypernym relations extracted from the string comparison module. Gold Standards taxonomies are used to evaluate the system's taxonomies for each domain. Our system achieved significant results across each domain.

Keywords*: Hyponym-Hypernym Relations, Taxonomy Induction, Word-Embedding*

1. Introduction

With the rapid growth of information and arrival of the knowledge economy that increase the dependency on the effective utilization of available vast information, it becomes vital to introduce a technique and tool that organize and retrieve relevant information efficiently. Presenting useful information in a well-ordered structure prevents overloading of information on its access and making the information access qualitative. Moreover, it's also incorporate the discovery and extraction of critical and relevant knowledge information [7]. With the passage of time information are increasing not in length but also in its meaning or in context, i.e. in order to understand the meaning of a particular word we need to understand its context first or its meaning in a particular context.

Taxonomy is a knowledge management tool that organizes wide information in a well hierarchical structure that mainly concerns on the access and extraction of right and relevant information. As organizing information in a structured form is not a new trend as it always been in everyone's concern according to [7].

Manual taxonomy construction is a time consuming and knowledge incentive process. Therefore, a lot of researches have been conducted to automate the process of taxonomy induction and to find hyponym-hypernym relations automatically. However, finding hyponym-hypernym relations automatically from unstructured text is a great challenge in the research area.

The main task is to use terms lists from different domains, to discover and fetch the hierarchical relations from a large corpus and then structure these extracted relations in hierarchical order.

Existing broad coverage lexical resources such as WordNet [17] and EuroWordNet [33] are extensively used for NLP tasks such as sentiment analysis, information retrieval etc. but faces a number of limitations. As these resources are costly and required a large amount of effort for it's creation and maintenance. Moreover, it is very cumbersome to add new word in the existing

resource. These existed manually crafted lexical resources includes more general vocabulary as compared to domain-specific vocabulary that's why they are incomplete and difficult to extend with new words. In these lexical resources words are semantically associated with each other. Number of research have been conducted to learn how to induce taxonomy automatically from unstructured text instead of creating time consuming manual taxonomy. The summary on the taxonomy learning from the text is given in [1].In this paper, we mainly concern with the identification and extraction of hypernym-hyponym relations from a large corpus by using a domain specific term list. Then, organize these relations in a hierarchical order to construct taxonomy. The domain-specific term list contains a noun and noun phrases.

In literature, different heuristics have been proposed that extract semantic relations from the text and construct taxonomy such as coordination and co-occurrence information [20], [34], [27], [35]. In lexical semantic resources such as WordNet, the terms are associated by a number of relations such as is-a (hypernym-hyponym), is-like(synonymy) [14] and meronymy (has-a or contain) relations [8] etc.

But in this research, we only focus the automatic identification and extraction of hypernym-hyponym relations from the text and arranging them in hierarchical taxonomies. Hypernym-hyponym relations are asymmetric and transitive but not the symmetric [17], [11].

Previous studies reveal that these automatic taxonomies can be used to extend the existing manual crafted semantic ontologies such as WordNet [29], [27] and Euro Word [31].

Our focus is to introduce a system that detects and extract hyponym-hypernym relations from a large text and structure them automatically in a hierarchical tree i.e. taxonomy. These large corpus incorporate a variety of general and domain specialized terms expressing different relationships between terms such as"type of"etc. However, most of these domain specialized terms are not available in manually created inventories such as WordNet.

The structure of the paper is as follow: In section 2 we provide the brief introduction of the state-of-art approaches, in section 3 we propose and explain our system. Section 4 we evaluate our system against Gold Standards and compare with existing systems. Finally, we conclude our system with a brief conclusion.

2. Related work

The main steps for semantic taxonomies construction include term extraction, relation extraction, and taxonomy construction. These terms can be provided by extracting them from existing manually crafted lexical resources or can be extracted from a large text.

Sate of art approaches for automatic taxonomy inductions incorporates pattern-based approaches, distributional approaches, graph-based approaches, sub-string inclusion method and clustering-based approaches.

Pattern-based approaches also known as rule-based approaches, the most common technique for hyponym-hypernym identification. It is highly inspired by Hearst [11]. She created few lexico-syntactic patterns manually and then traverse the large text in a particular manner if these patterns are found then a system incur hyponym-hypernym relation between terms. Number of existing approaches based on these Hearst-style patterns [4], [23], [25], [27], [16], [18].

In [22], semantic relations are identified between terms based on generic patterns and then pruning methods are applied to get more accurate relations. Most of the studies used the relations extracted from pattern-based approaches as an input to post-processing methods for results improvement such as [26] improved results with Support Vector Machines. The acyclic graph is used to build a generic pattern-based algorithm based on web-documents to extract hypernyms in [19].

In most of the previous studies such as [2] combined approaches are used to identify and extract most relevant hypernym relation but in a way that output from one system is an input to another system. Moreover, they also used morphosyntactic analyzer approach [12].

One of the conventional approaches is sub-string inclusion method. It compares the string representation of two terms. If one term is contained in another term then the contained term is considered as hypernym. Such as computer science is a hyponym of science, where science is the

head-modifier. The morphosyntactic analyzer approach is also based on this approach but includes more working principles. First, it looks the single word, if its suffix is the existing term then it considered as hypernym. Secondly, if the word is a multi-word term contained part is a hypernym similar to sub-string inclusion method. Finally, if preposition exists between two nouns then the first noun is a hypernym such as sociology is a hypernym of the sociology of culture.

Another approach [31], which first search patterns from a large corpus and then utilized these patterns to enhance the efficiency of the system like question answering system. Semantically similar words are those who occur in a similar context [10]. For target word, word context can be defined by using the information related to syntactic and co-occurrence of the surrounding words.

In clustering-based approaches, terms are clustered by utilizing their word context in a particular corpus. Then, labels are assigned to each cluster. Hyponym-hypernym relations can be found between cluster' label and cluster' members. Authors in [3], [23], created noun clusters based on syntactic

the dependency on words.

Recently, with the advent of the neural network, the distributional approaches have gained a lot of fame and become a popular tool for knowledge rich problems. According to the research [13], in distributional space, the hyponym includes a number of the distributional features of their hypernyms.

Distributional semantics embeds the semantic knowledge of words in vectors in a vector space. The closeness of these vectors in vector space infers the semantic similarity of the associated terms. This approach is very useful to find the semantic similarity of the word even if they don't occur explicitly in the corpus. However, it is difficult to identify asymmetric relations between concepts.

3. System description

The proposed approach is simple and domain independent that is able to process large data along with automatically extracting asymmetrical hierarchical relations i.e. hyponym-hypernym relations from unstructured text and subsequent taxonomy construction. The system takes a list of domain-specific terms and large corpus to train vector space model and then generates a well-structured taxonomy. It includes four main steps. First, the systems crawls the general corpus and generate the vectors of 300 dimensions for each domain-specific term. Then, the system finds and extracts the hypernyms by using the generic algorithm based on distributional semantics calculations. In the last step, the system adds hyponym-hypernym relations extracted from string comparison module.

3.1. Vector space model

Word2vec is one of the vector space model, used to convert text in a numerical form that deep neural network can understand. Word2vec vectorizes about words, the vectors are known as word-embedding. These vectors are used to describe one another but are totally different. These vectors are not only making text to machine-readable but also used to find the similarities between terms in a given corpus.

Word2vec generates vectors like autoencoder but trains words against neighbor terms in a corpus rather than train against input term as in Boltzmann machine [6]. Word2vec generates vectors based on two methods, continuous bag of words (CBOW) and skip-gram [9], [28]. In CBOW context is utilized to predicts target word, where multiple words are used to represents context. The contexts of the word can define as the vectors of the words those occur near to that word in a large corpus. However, the working of skip-gram is reverse to the working of CBOW. It takes words as an input and predicts target context. For implementation, we used a skip-gram model that generates more accurate results on large corpus. These vectors can be adjusted to predict word's context more accurately.

The word2vec not only make more accurate guesses about a word's meaning based on their past occurrence but also establish associations between terms such as man-boy ≈women-girl and can perform clustering of documents and classify them topic wise. These clusters can use in recommendation systems, such as e-commerce and scientific research etc. and in sentiment analysis. Word2vec plays a crucial role in grouping the similar words in a vector space.

These vectors are mathematical representations of words context. It automatically generates vectors without human involvement. Word2vec not only used for sentence parsing but is also applied to multiple things such as graphs of social media, playlists, codes especially where patterns are involved. The output of word2vec is like vocabulary in which multiple vectors are linked with word these can be input to deep learning-net or can query for relations discovery between words.

3.2. Training corpus

For the implementation purpose, we trained word2vec model over GoogleNews-vectors-negative-300 model corpus. This corpus has about 100 billion words. For each domain-specific term, word2vec generates vectors of 300 dimensions.

The assumption of proposed methodology is similar to our previous system [36]. Our assumption is, "if two words are hyponyms of a third one, then the distance from each of this two word to that hypernym should be similar" [36].

3.3. Taxonomy induction from word-embedding

1. First, we generate vector from word2vec model of each word of a domain-specific list.

- Use vectors of each word, available in the word2vec model.
- For compound terms, if one or more of its parts available in a model then we add up all the available vectors to get the desired vector.
- Ignore the term if it is not available in a model.

2. Then, we create a distance matrix that stores the computed cosine distance of each pair of term's vectors. Let Animal, Dog, Cat, Car, and Shark be the terms:

	Animal	**Dog**	**Cat**	**Car**	**Shark**
Animal	d0	d1	d2	d3	d4
Dog	d5	d6	d7	d8	d9
Cat	d10	d11	d12	d13	d14
Car	d15	d16	d17	d18	d19
Shark	d20	d21	d22	d23	d24

3. For each row of a distance matrix:

- First, sort the row.
- Then, compute the difference between two adjacent values in a row.
- By applying the constraints that the difference must be greater and equal to zero. Because we already sort the row that makes the situation to subtract the smaller value from larger value and the other reason is as a delta is a difference between the two distances that must be positive as distance always positive.
- d1=(Animal, Dog)
- d2=(Animal, Cat)
- Delta= d2-d1
- Delta >= 0
- In this way, we considered all possible valid triples with the help of computed difference from the distance matrix.

- Triples→ {Animal,Dog,Cat},{Animal,Cat,Shark},{Animal,Cat,Car},……

4. In this step, we compare triples by taking the mean of distances between terms.

- d1=(Animal, Dog)
- d2=(Animal, Cat)
- d3=(Dog, Cat)
- mean=(d1,d2,d3)

5. Then, all triples are sorted according to the computed mean values.

- 0.1→ {Animal,Dog,Cat}
- 0.2→ {Animal,Shark,Cat}

6. Verify each triple before adding triple to the hierarchy.
7. In relation verification step, three steps are included:

- If the hyponym is in taxonomy as a child, it means it is already associated with parent(hypernym).
- If the particular relation is already in taxonomy,then, we do not add it. As this constraint avoid the duplication of the relations.
- If the particular relation is not found in taxonomy, then we store hyponym in a separate list in each iteration.
- For each relation, we checked either hypernym is available as hyponym. If yes we can easily locate the hierarchy of hypernym.

8. After verification step, all sorted triples are added to the hierarchy.

The runtime complexity of the proposed algorithm is $O(n^2 * \log n)$. As the $O(n^2)$ triples are formed and sorted by taking the average cosine distance of each triple. Figure 1 shows the overall working of the proposed algorithm Figure 2 demonstrates how the relations are verified before adding to the taxonomy.

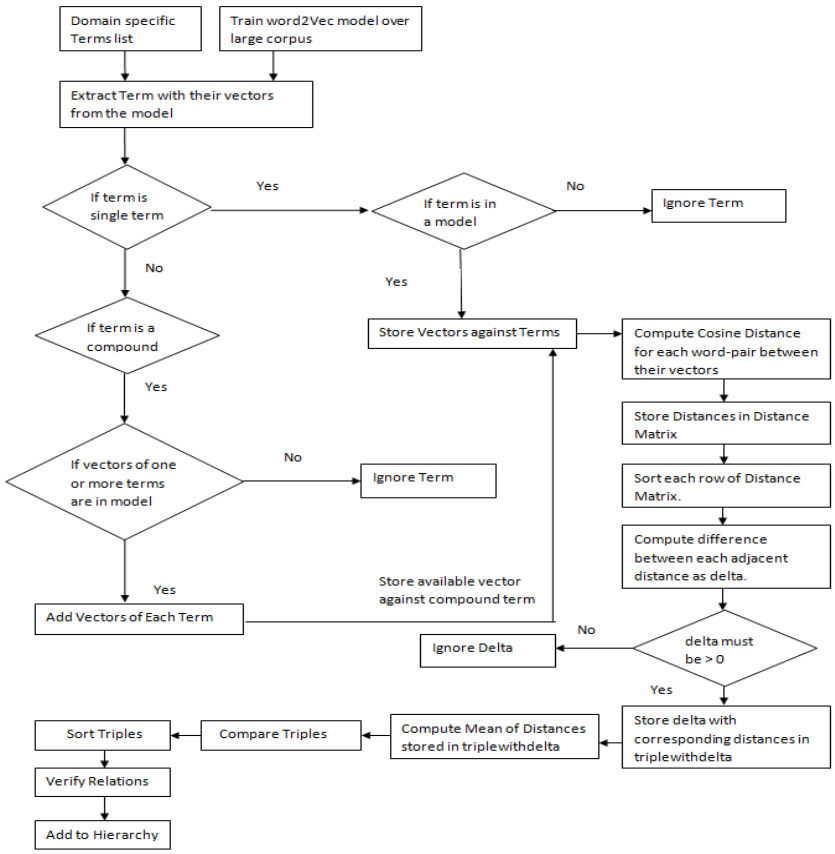

Figure 1. Flow Diagram of Proposed Algorithm

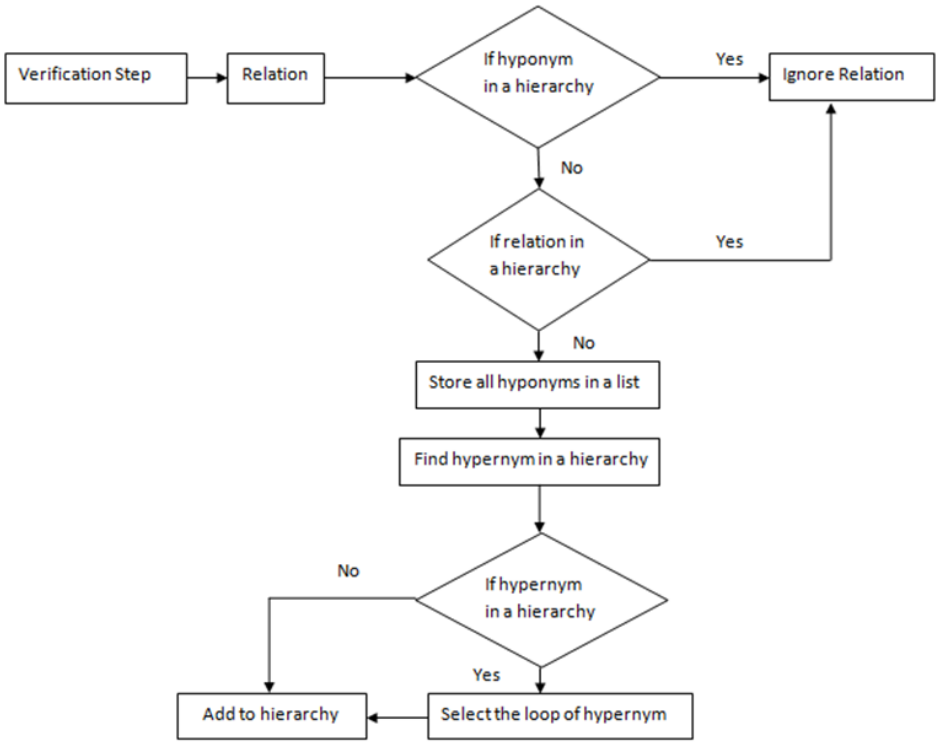

Figure 2. Flow Process of Relations Verification

3.4. String Comparison Module

The string comparison module is used to generate a list of hyponym-hypernym relations in a way similar to [12]. It's working principles are, if one term contains another term then the contained term is hypernym. For example, "computer science" is a "science". Secondly, if "of" preposition occur between two noun terms then the first term is considered as hypernym. Finally, we set the constraint that length of hypernym should be greater than three for valid hypernym detection.

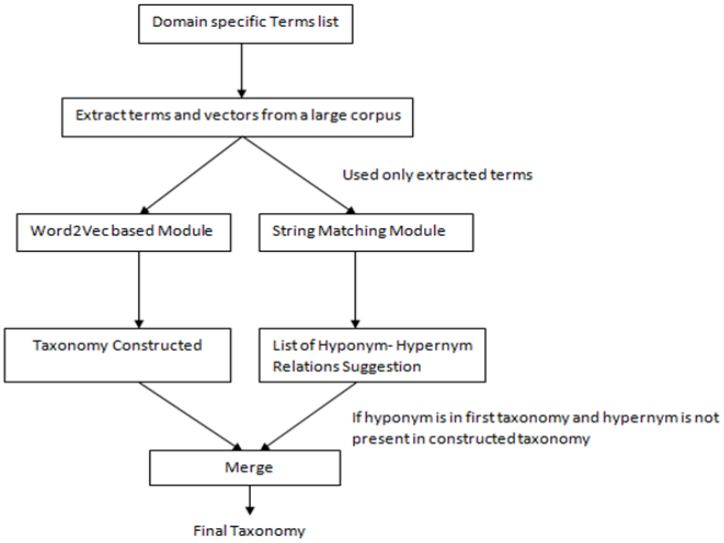

Figure 3. System Diagram of the Proposed System

We include only those hyponym-hypernym relations from the string-comparison method if hyponym is available in a first taxonomy. Secondly, for that particular hyponym, the hypernym is not available. This is to avoid the duplication of relations and prevents the addition of inverse relations. Figure 3 shows the systematic diagram of the proposed system.

4. Evaluation

The Evaluation of taxonomies is not a trivial task as it incorporates two steps. First each relation in the system taxonomies and in the gold standard taxonomies are compared. In this step it is important to check that relation should exist between two correct terms and in correct direction. In the second step of evaluation, the hierarchical structure of system taxonomies is compared with the hierarchical structure of the gold standard taxonomies, in order to verify that the relations in system taxonomies are in correct order as in gold standard taxonomies.

A number of measures required to perform in order to evaluate the quality of constructed taxonomies. We evaluate the system taxonomies by using benchmarks provided by SemEvalTexEval-2 workshop. The system generated six taxonomies for three different domains i.e. food, science, and environment. For the evaluation of each taxonomy six benchmarks are considered. Each benchmark contains a list of terms that are extracted from existing manually crafted lexical resources such as WordNet, Eurovoc etc. and the Gold Standards. The jar file written in java is used to compare the system taxonomies against the Gold Standards.

In this paper, we evaluate the system performance and quality in term of F-score and (F&M) measures. In order to compute F-score, first, we need to calculate recall and precision of the system taxonomies against Gold Standards. The recall is the number of common edges of the system taxonomies with the Gold standards divided by a total number of edges in the Gold Standards. However, the precision is the number of common edges of the system taxonomies with the Gold standards divided by a total number of edges in the system taxonomies. Moreover, (F&M) is the Cumulative Fowlkes & Mallows Measure [32], it computes the similarity of hierarchical clusters of system taxonomies against the Gold Standards.

4.1. Results

In this section, we present our system's results when compared to Gold Standards for each domain as shown in Table 1. The results are significant across all the domains. For the science related benchmarks, the system performs slightly better. However, f-score for environment benchmark and (F&M) measure for the food benchmark are low as compared to other benchmarks. The reason for low f-score and (F&M) is either the particular domain-specific term is not present in the large corpus or the term is not frequently available. For infrequent terms, the system cannot determine their context properly and the generated word-embedding of these terms are useless.

Moreover, our system is not utilizing any existing lexical resource. Furthermore, the limitations of the system are: if a term is not available in the training corpus, the word-embedding of that particular term will not be generated. The word-embedding of the infrequent terms will not be considered. The selected hypernym in the system taxonomies should also be present in the domain-specific term lists. But still, the results of our systems are very close to the existing best system.

Table 1. Results Summary against Gold Standards

Measures	Environment (Eurovo)	Food (WordNet)	Food	Science (Eurovo)	Science	Science (WordNet)
Precision	0.196	0.2042	0.1802	0.268	0.2475	0.3121
Recall	0.2644	0.2746	0.206	0.3029	0.3226	0.3306
F-score	0.2251	0.2342	0.3862	0.3049	0.2801	0.3211
F&M	0.2264	0.2342	0.1922	0.296	0.4498	0.3079

We also compared our system results with the other systems of the SemEvalTexEval-2 workshop i.e. TAXI, NUIG-UNLP, QASSIT, JUNLP and USAAR in Table 2. TAXI [21] system based on Hearst-patterns and sub-string inclusion. For hyponym-hypernym relations identification and taxonomy induction, they used both general and domain-specific corpora. USAAR [30] proposed Hyponym Endocentricity system based on the assumption that compound terms are endocentric in nature, means one part of the compound term represents the whole compound term. NUIG-UNLP [24] utilized word-embedding generated from the glove word space model. The model is trained over corpus extracted from Wikipedia dump. The system identifies hypernym by adding the average vector offset to the corresponding hyponym vector. For average vector offset, they compute an average of the difference between a vector of hypernym and vector of hyponym in a vector space for 200 pairs of words. The JUNLP [15] system for taxonomy induction relied on the lexical resource, i.e. BabelNet and sub-string inclusion method. The QASSIT [5] system extracted lexical patterns and applied pre-topological space graph optimization techniques based on genetic algorithm.

A table 2 show, the proposed system is better than the current system on vector space model. However, it is close to TAXI, who ranked first among the systems. Table 1 demonstrates that terms from science field are more frequently occur in the corpus than the terms from food domain.

Table 2. Results Comparison with other Systems

Measures	USAAR	TAXI	JUNLP	QASSIT	NUIG	SYSTEM
Precision	0.5677	0.3318	0.1474	0.1858	0.2200	0.2347
Recall	0.1916	0.3174	0.3009	0.263	0.1100	0.2835
F-score	0.259	0.3212	0.1961	0.2176	0.2028	0.2919
F&M	0.0013	0.2908	0.1498	0.4064	0.0410	0.2844

5. Conclusion and future work

In this paper, we presented a system based on vector space model and string comparison method for automatic hyponym-hypernym identification from domain-specific term lists and structuring them in taxonomy. We evaluate our system on the dataset provided by the SemEvalTexEval-2 workshop on three different domains. Our system shows significant performance against each domain without utilizing any existing lexical resource.

In future, we plan to train neural network from a number of domain-specific corpora. We also plan to incorporate wiki miner results for taxonomy induction.

6. References

[1] C. Biemann. "Ontology learning from text: A survey of methods. In LDV forum", volume 20, pages 75–93, 2005.

[2] W. Bosma and P. Vossen. Bootstrapping language neutral term extraction. In LREC, 2010.

[3] S. A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from the text. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pages 120–126. Association for Computational Linguistics, 1999.

[4] S. Cederberg and D. Widdows. Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, pages 111–118. Association for Computational Linguistics, 2003.

[5] G. Cleuziou and J. G. Moreno. Qassit at semeval-2016 task 13: On the integration of semantic vectors in pretopological spaces for lexical taxonomy acquisition. Proceedings of SemEval, pages 1315–1319, 2016.

[6] A. Fischer and C. Igel. An introduction to restricted Boltzmann machines. In Iberoamerican Congress on Pattern Recognition, pages 14–36. Springer, 2012.

[7] A. Gilchrist, P. Kibby, and B. Mahon. Taxonomies for business: access and connectivity in a wired world.TFPL London, 2000.

[8] R. Girju, A. Badulescu, and D. Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 1–8.Association for Computational Linguistics, 2003.

[9] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722, 2014.

[10] Z. Harris. Mathematical structures of language. 1968.M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th conference on Computational linguistics-Volume 2, pages 539–545. Association for Computational Linguistics, 1992.

[11] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th conference on Computational linguistics-Volume 2,pages 539–545. Association for Computational Linguistics, 1992.

[12] E. Lefever. Lt3: a multi-modular approach to automatic taxonomy construction. In 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 944–948. Association for Computational Linguistics, 2015.

[13] A. Lenci and G. Benotto. Identifying hypernyms in distributional semantic spaces. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 75–79. Association for Computational Linguistics, 2012.

[14] D. Lin, S. Zhao, L. Qin, and M. Zhou. Identifying synonyms among distributionally similar words. In IJCAI, volume 3, pages 1492–1493, 2003.

[15] P. Maitra, D. Das, and I. Kolkata. Junlp at semeval-2016 task 13: A language independent approach for hypernym identification. Proceedings of SemEval, pages 1310–1314, 2016.

[16] V. Malaisé, P. Zweigenbaum, and B. Bachimont. Detecting semantic relations between terms in definitions. In COLING, pages 55–62, 2004.

[17] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. International journal of lexicography, 3(4):235–244, 1990.

[18] V. B. Mititelu. Hyponymy patterns. In International Conference on Text, Speech and Dialogue, pages 37–44. Springer, 2008.

[19] R. Navigli and P. Velardi. Learning word-class lattices for definition and hypernym extraction. In Proceedings of the 48th annual meeting of the association for computational linguistics, pages 1318–1327. Association for Computational Linguistics, 2010.

[20] J.-H. Oh, K. Uchimoto, and K. Torisawa. Bilingual co-training for monolingual hyponymy-relation acquisition. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, pages 432–440. Association for Computational Linguistics, 2009.

[21] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. Proceedings of SemEval, pages 1320–1327, 2016.

[22] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Proceedings of the 21st International Conference on

Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 113–120. Association for Computational Linguistics, 2006.

[23] P. Pantel and D. Ravichandran. Automatically labeling semantic classes. In HLT-NAACL, volume 3, pages 3–2, 2004.

[24] J. Pocostales. Nuig-unlp at semeval-2016 task 13: A simple word embedding-based approach for taxonomy extraction. Proceedings of SemEval, pages 1298–1302, 2016.

[25] E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. arXiv preprint cmp-lg/9706013, 1997.

[26] A. Ritter, S. Soderland, and O. Etzioni. What is this, anyway: Automatic hypernym discovery. In AAAI Spring Symposium: Learning by Reading and Learning to Read, pages 88–93, 2009.

[27] B. Roark and E. Charniak. Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In Proceedings of the 17th international conference on Computational linguistics-Volume 2, pages 1110–1116. Association for Computational Linguistics, 1998.

[28] X. Rong. word2vec parameter learning explained. arXiv preprint arXiv:1411.2738, 2014.

[29] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 801–808. Association for Computational Linguistics, 2006.

[30] L. Tan, F. Bond, and J. van Genabith. Usaar at semeval-2016 task 13: Hyponym endocentricity. Proceedings of SemEval, pages 1303–1309, 2016.

[31] L. Van Der Plas, G. Bouma, and J. Mur. Automatic acquisition of lexico-semantic knowledge for qa. In Proceedings of the IJCNLP workshop on Ontologies and Lexical Resources, pages 76–84, 2005.

[32] P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. Computational Linguistics, 39(3):665–707, 2013.

[33] P. Vossen. A multilingual database with lexical semantic networks. Springer, 1998.

[34] D. Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 197–204. Association for Computational Linguistics, 2003.

[35] D. Widdows and B. Dorow. A graph model for unsupervised lexical acquisition. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics, 2002.

[36] Zafar, Bushra, Michael Cochez, and Usman Qamar. "Using Distributional Semantics for Automatic Taxonomy Induction." *Frontiers of Information Technology (FIT), 2016 International Conference on*. IEEE, 2016.