

Using Distributional Semantics for Automatic Taxonomy Induction

Bushra Zafar
College of Electrical &
Mechanical Engineering,
National University of
Science and Technology
Islamabad, Pakistan
bushra.zafar025@gmail.com

Michael Cochez
Fraunhofer – FIT
Sankt Augustin, Germany
michael.cochez@fit.fraunhofer.de

Informatik 5
RWTH Aachen University
DE-52056 Aachen, Germany

Usman Qamar
College of Electrical &
Mechanical Engineering,
National University of
Science and Technology
Islamabad, Pakistan
usmanq@ceme.nust.edu.pk

Abstract—Semantic taxonomies are powerful tools that provide structured knowledge to Natural Language Processing (NLP), Information Retrieval (IR), and general Artificial Intelligence (AI) systems. These taxonomies are extensively used for solving knowledge rich problems such as textual entailment and question answering. In this paper, we present a taxonomy induction system and evaluate it using the benchmarks provided in the Taxonomy Extraction Evaluation (TExEval2) Task. The task is to identify hyponym-hypernym relations and to construct a taxonomy from a given domain specific list. Our approach is based on a word embedding, trained from a large corpus and string-matching approaches. The overall approach is semi-supervised. We propose a generic algorithm that utilizes the vectors from the embedding effectively, to identify hyponym-hypernym relations and to induce the taxonomy. The system generated taxonomies on English language for three different domains (environment, food and science) which are evaluated against gold standard taxonomies. The system achieved good results for hyponym-hypernym identification and taxonomy induction, especially when compared to other tools using similar background knowledge.

I. INTRODUCTION

Semantic taxonomies are powerful tools that provide structured knowledge to Natural Language Processing (NLP), Information Retrieval (IR), and general Artificial Intelligence (AI) systems. Semantic taxonomies are extensively used for solving knowledge-rich-problems such as textual entailment [1] and question answering [2]. There are a number of manually created and curated taxonomies such as YAGO [3], Freebase [4], SUMO [5], etc. These taxonomies have excellent quality and cover a broad domain.

However, manually curating a taxonomy incurs a high cost as can be seen in taxonomies like WordNet [6] and more encompassing systems like Cyc [7]. Moreover, these systems remain incomplete and it is sometimes difficult to add new terms because of contradictions. Further, customization for different domains can be cumbersome. Nowadays, partially due to rapid technological development and availability of data sources, it has become feasible to construct semi-automatic, or even automatic, domain specific taxonomies, instead of creating those manually. [8] The process of taxonomy construction goes by many names like *automatic taxonomy induction*,

ontology learning, *semantic relation classification*, and so on [9]. A number of different theories and methodologies have been proposed to induce taxonomy automatically. Very common are pattern-based approaches. The working principle of these approaches is to define patterns or rules first and find matching text in a large textual corpus. Often a form of bootstrapping is used. The system iteratively learns more patterns from the corpus by deriving patterns from the results found in the previous round. In this paper we focus on a distributional semantics approach. We utilize the similarity of the context in which words occur to infer the meaning of the words. The context of a word is usually defined as the set of words that often occur near in a large corpus. We are also somewhat inspired by clustering approaches, which incorporate clustering based on word (or context) similarity.

In this paper we present a system which creates a taxonomy from a given set of words. This way, we can compare to the results of the SemEval – Taxonomy Extraction Evaluation (TExEval-2) workshop¹. The presented framework is semi-supervised and combines the strength of word-embedding, clustering, and string-matching to identify hyponym-hypernym relations.

The remainder of the paper is structured as follows. In the next section we provide an overview of related work. Then, in section III we introduce our own system. In section IV we present our evaluation and compare the results with other systems. Then we finish with a short conclusion.

II. RELATED WORK

The process of automatic taxonomy construction, can be split in two steps. First, one needs to obtain the terms which have to be in the taxonomy. These can either be provided or have to be identified and extract from a text corpus. Second, these terms have to be placed in a hierarchy, forming the taxonomy. Often these two steps are combined into one as in the pattern-based approaches which we will review first. After that we will look into clustering, classification, graph,

¹<http://alt.qcri.org/semEval2016/task13/>

and distributional approaches. Note that these approaches are often combined, like for example Kozareva and Hovy [10] who made a system based on patterns and graph algorithms for creating the taxonomy.

Pattern-based (also sometimes called rule-based) approaches form perhaps the most common class of automatic taxonomy induction tools. The core idea of these approaches is to establish certain patterns and if these patterns are found in the corpus the system infers a hypernym–hyponym relation between them. For example, if the pattern a $[[\text{noun1}]]$ is a $[[\text{noun2}]]$ (where the double rectangular braced words are placeholders for actual words in the text) is found, then one can conclude that *noun2* is a hypernym of *noun1*. If these patterns are chosen wisely, these systems achieve a high accuracy and often remain relatively simple in implementation.

Obviously, this kind of patterns can be used for applications beyond hyponym–hypernym detection. Also other relations like part-of (meronymy), is-like (synonym), is born in, etc. can be detected using patterns.

The creation and selection of patterns can be done either manually [11] or automatically using automatic bootstrapping [12]. The bootstrapping technique starts from a limited set of man-made patterns and extracts words fitting the patterns from the corpus. Then, these extracted words are used to find new rules from the corpus. If we, for example, using the rule above, learned that a cake is a dessert, then we can now go through the corpus and detect other sentences containing cake and dessert. If we find the sentence “cake is my favorite dessert”, we could instantiate the new rule $[[\text{noun1}]]$ is my favorite $[[\text{noun2}]]$. Now, this new rule can be used to find more pairs of words and the process gets repeated.

Already in early works pattern-based approach are used for finding is-a relation from a large corpus. Hearst [12], for the first time, utilized a set of hyponym-hypernym patterns and performed bootstrapping to automatically find the is-a relation from a corpus. These patterns are now known as Hearst-style patterns. Others have also used this pattern style like, for instance, Mann [13] who used Hearst-style patterns to extract the is-a relation for proper nouns.

However, a drawback of pattern-based approaches is the coverage of diverse patterns for varying corpora, as these patterns are unavailable or have only limited occurrences in a given corpus. Previous research did observe that a large corpus increases accuracy of the pattern-based approach. [14], [11] Another problem with this approach (which also exists in some other approaches) is that concepts are extracted in pairs (sometimes triples) and then connected to form a taxonomy. This may lead to inconsistent chains where a word is recursively a hypernym of itself.

Another way to find concepts with a hyponym-hypernym relation is to compare their string representations with each other. For example, the concept *fish soup* is a hyponym of *soup*. In the string representation we see that the word *soup* has what Jones [15] called a *head-modifier*. Also other modifiers have been studied in the literature. Lefever [8] also created methods, for finding the relation we are studying,

using morpho-syntactic analyzes. She essentially proposed three identifying principles. First, in single word terms the suffix is considered a hypernym if the prefix is an existing term. For example, *pin* is a hypernym of *candlepin*. Second, for multi word terms, a term t_1 is hypernym of multi word term t_2 if t_2 contains t_1 (as pioneered by Jones). Finally, prepositions in words are used. For example, the concept *sociology of culture* contains the preposition *of* and hence we would decide that *sociology* is a hypernym of *culture* (example from [8]). There are a few limitations associated with string-based methods. The first one is that it can only be applied to compound terms (e.g., the hypernym of *chemistry* cannot be found using a string-based method). The second one is that a string-based method cannot differentiate direct and indirect hypernyms. For instance, *animal science* is a *life science*, which in turn is a *science*. However, a string-based method will identify *science* as the hypernym of both.

Another large group of approaches are based on clustering. For these approaches, word contexts are represented as vectors containing the values for selected features. Next, a similarity between the vectors is defined and then the clustering is performed [16], [17]. A number of features have been used to create these vectors such as noun-verb relations [18], co-occurrence [19], and so on. Using clustering-based approaches it is possible to identify relations which do not occur explicitly in the text corpus. However, labeling of clusters is required in clustering-based approaches, which is difficult for taxonomy induction and evaluation. Moreover, similarly to pattern-based, the clustering-based approaches fail to produce coherent clusters for small corpora, and hence do not produce good results. To our knowledge, clustering-based approaches have only been used to identify and extract hyponym-hypernym (is-a) and synonym (is-like) relations.

Snow et al. [20] used sentences and parse trees for hypernym identification. First, the sentences are extracted and if they contain two terms parse trees are used to extract patterns. These patterns were consecutively used to train the hypernym classifier to induce a taxonomy from text corpora. Yang and Callan [9] introduced a semi-supervised approach for taxonomy induction that merges syntactic dependencies, lexical- syntactic patterns, co-occurrence and an ontology metric. The ontology metric is storing the distance between the terms and then the terms are clustered using these distances. Veraldi et al, [21] utilized a branching algorithm to induce taxonomies based on definitions extracted from the Web and a large corpus.

The graph-based approaches (e.g, [22], [23]) are based on both lexical similarity and statistical co-occurrence to automate taxonomy induction. These models utilize a large corpus to extract topical terms and find relationships between them. Then, the model creates a weighted-graph that represents topics and their sub-ordinates and applies a graph partitioning algorithm. This process is applied iteratively and a topic-dependent taxonomy is formed.

Recently, distributional or word embedding approaches have received attention of the research community [24], [25], [26].

This is mainly due to the recent uptake and practical usability of deep learning and neural networks. The general idea is that semantic knowledge of terms can be embedded in a vector space. In this space each term is represented by a single vector and the closeness of vectors represents the semantic similarity of the associated terms. Or in other words, the terms are distributed across the vector space in such a way that similar words are grouped together or found nearer to each other [27]. Due to the rapid advancement in neural nets and word embedding approaches [28], [29], [30], [31], the vector space model has become a focus point and a popular tool for relations extraction and taxonomy induction problems [32], [33], [34]. Pocostales [35] used vector offset for hyponym-hypernym identification despite the fact that Rei and Briscoe [36] observed that this is difficult because the relation is asymmetric. Also in our system, which will be presented in the next section, we make use of a vector space model, but we do use the vectors differently.

III. PROPOSED TAXONOMY INDUCTION METHOD

The method we propose in this paper combines the ideas of word embedding and the string-based methods of Lefever [8]. In the next subsections we present the word embedding model we used, the way we exploited the vectors representation to find hyponym-hypernym relations, and how we augmented this with the string-based methods.

A. Word2Vec

One recent, successful approach to word embedding is Word2Vec [37]. Word2vec is a predictive model and two-layer neural network that takes raw text as an input and generates a word embedding. This embedding provides a vector for each word in the corpus. These vectors retain the semantic meaning of the words as we will discuss further below. Neural networks are often used to identify patterns but require training data. The training data must be presented to the network as numerical vectors. A variety of data types, like e.g., images, sound, and text, have been converted to numerical vectors and subsequently used as an input of a neural network.

Neural networks are nearly always tweaked for the application at hand. In the case of word2vec hierarchical softmax and negative sampling are used to train the network (For a deeper explanation see [38]). Word2vec implements two methods for vector generations, continuous bag of words (CBOW) and skip-gram [38], [39]. CBOW uses the context for prediction of the target word, whereas the skip-gram utilizes words to predict the target context. Due to its high accuracy and success in preliminary experimentation, we used skip-gram for model generation.

For our application, we are mainly interested in the vectors which word2vec produces. For each word in the training data set, the neural network can produce a vector which encodes the semantic meaning of the word. Words which have a closely related semantic meaning, get mapped to vectors with a small cosine distance (or angle) between them. Note that it is possible that a model does not include all terms we are

interested in. In the description of the algorithm below we will describe how we solved that problem.

B. Offset Model for hyponym and hypernym identification

Word-embeddings generated using distributional approaches, store some specific semantic and syntactic knowledge in the created vectors [40], [30]. For example, it has been observed that word2vec vectors representing capitals have an interesting relation to the vector representing their country. Namely, that if one subtracts the capital vector from its country vector, then the remaining vector will, for all capital-country pairs, be very similar. Thus, one can conclude that this remaining vector encodes the ‘is-capital-of’ semantics. To give an example, we can take Germany and France, with their respective capitals. Using their vector representation, we will notice that $v_{Paris} - v_{France} \approx v_{Berlin} - v_{Germany}$, which can also be used to discover the capital of a country because it is also true that $v_{Paris} \approx v_{Berlin} - v_{Germany} + v_{France}$.

A further property of the vector space model is that vectors representing words with a similar meaning, have a small (cosine) distance to each other. The cosine distance is defined as $cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$. Further, because of the phenomenon described above for the is-capital-of relation, we make the assumption that if two words are hyponyms of a third one, then the distance from each of these two word to that hyponym should be similar. Using this assumption, we created the following algorithm for finding hyponym-hypernym relations between words.

- 1) For each word, we need a vector from the word2vec model. If a term is in the model, we use its vector directly. If a compound term has one or more of its words in the model, we use the sum of these vectors. However, if none of the parts is in the model, we drop the term and do not consider it further.
- 2) For each pair of terms, compute the cosine distance between their representative vectors, creating a distance matrix.
- 3) For each (unordered) triple of terms, compute the harmonic average of cosine distances computed in step 1. Sort the triples according to this average in ascending order.
- 4) Now, we use the assumption which we explained above. We iterate over the triples (looking at the one with lowest average distance first) and attempt to find out which of these three terms should be a hypernym if the other two are its hyponyms. According to our assumption, the hypernym must be the word which causes the distances, from the two hyponyms to itself, to be as close as possible. Hence, we try the three possible orderings and select the one which minimizes the difference between the two distances.
- 5) In each triple from the previous step we have now identified one hypernym and two hyponyms. These can be seen as part of a partial ordering. However, cycles in the ordering created this way can occur,

and we have to eliminate those. Note that we sorted the triples at an earlier stage. We will now prioritize triples with a smaller average distance when building the hierarchy. Essentially, we build the hierarchy iteratively and whenever information provided by a triple which comes later in the process contradicts with an earlier triple, we ignore the new information. The idea is that triples with a smaller average distance are more likely to contribute correct information to the hierarchy. The runtime complexity of this algorithm is $O(n^3 \log n)$. The reason is that $O(n^3)$ triples are formed and these are then sorted according to the average cosine distance as described above.

Above we described the process we used for constructing the taxonomies, in the next section we will show the results of our empirical evaluation.

IV. EVALUATION

In this section, we evaluate the proposed taxonomy creation tool. First, we describe the benchmarks against which we obtained the performance of the proposed system. Then, we present our results and compare them with the outcome of other systems.

A. Evaluation Method

To evaluate our approach, we use the benchmarks provided by the SemEval–TexEval-2 workshop². Each benchmark consists of a list of terms which are to be placed in a taxonomy and a gold standard taxonomy. The goal of a system is to generate a taxonomy which as close as possible resembles the gold standard taxonomy, without using the gold standard nor the sources which were used to create it. Several benchmarks are provided coming from various domains to reduce the success of systems which are tailored to specific topics. The domains are environment (1 benchmark), food (2 benchmarks), and science (3 benchmarks).

In order to measure the quality of a produced taxonomy, several metrics are used. First, there is *precision* which measures the fraction of correctly found relations in the total number of relations identified by the system under scrutiny. It is a measure of how correct the results are. Second, *recall* measures the number of correctly found relations divided by the total number of relations in the gold standard. This is a measure of completeness. Precision and recall are in a sense competing. If one wants to get to high quality results, one can easily do so by giving up on completeness, i.e., only the results which are absolutely certain are retained. Also the opposite is true, one can increase recall by giving up on precision. To still keep systems comparable, the harmonic mean of the two, also called the F-score, is often used. Finally, the last measure is a cumulative measure [21], also known as *Cumulative Fowlkes & Mallows Measure* (F&M). This measures the similarity between two taxonomies by combining the similarities of hierarchical clusters.

²<http://alt.qcri.org/semeval2016/task13/>

B. Extending the Taxonomy with String-Matching Module

To augment the results obtained using the technique described in the previous section we use a string matching approach which is very similar to what was done in reference [8]. We made two minor changes: we only look for the *of* preposition and also make sure that the hypernym always contains at least three characters.

Note that the string matching technique creates a second taxonomy. We use that taxonomy to augment the word-embedding based one by including all hyponym-hypernym pairs the substring method finds, if the hyponym is already present as a hyponym in the word-embedding based taxonomy.

C. Results

In this subsection, we present the outcomes for these benchmarks and compare them with other systems. For our system, we used the pre-trained *GoogleNews-vectors-negative-300* model³. This model was trained using close to 100 billion words; and for 3 million words and phrases it contains a 300-dimensional vector.

In table I we present the total number of terms in the benchmark and how many of those are found (at least partially) in the trained model. The results for the above mentioned metrics across the 3 different domains can be found in table II. These results are for the proposed system and obtained by comparing it against the provided gold standard. From these measurements, it seems like our system performs fairly consistent. We notice, however, that it performs slightly better in the science related benchmarks and that the value for the F&M metric is very low for the WordNet food benchmark. The reason for this is that the latter benchmark contains many words which are not frequent in the corpus. These infrequent words have two problems: first, the system is unable to determine their context properly and second, if a word is not frequent enough it is known that the embedding is essentially useless and it is dropped from the embedding. For the other domains, i.e., science and environment, the evaluation measures are comparable to the best system's results. This shows that most of the terms from these domains are available in the corpus. However, we are not utilizing any supervised approach or lexical resource to improve the system's performance. Moreover, we limited our algorithm to consider only hypernyms in the domain-specific term list. Still, our outcome is close to the best system.

We provide the averages for the different metrics for the systems in the last TexEval workshop in table III. From a method perspective, the NUIG-UNLP system [35] is the closest to our proposed system. The system is based on vector offset in word embedding spaces (see also section III-B) combined with a string-based system. The workshop organizers also implemented a string-based method for identification of hypernym from compound terms, which forms the baseline approach. The other systems use a variety of methods to create a taxonomy. JUNLP [41] used a lexical resource (BabelNet)

³See <https://code.google.com/archive/p/word2vec/>

TABLE I

THE NUMBER OF TERMS IN THE EVALUATION SETS AND THE NUMBER OF THESE TERMS WHICH HAD AT LEAST ONE WORD IN THE WORD2VEC MODEL.

Domains	Terms in Domain list	Terms found in Corpus
Environment(Eurovoc)	261	260
Food	1555	1134
Food(Wordnet)	1486	1400
Science	456	442
Science(WordNet)	370	313
Science(Eurovoc)	124	122

for hyponym-hypernym relation extraction and string-based system. The QASSIT [42] system relies on lexical patterns and pre-topological space graph optimization techniques based on genetic algorithm for hyponym-hypernym identification and taxonomy induction. USAAR [43] proposed *Hyponym Endocentricity* which is similar to a string-based system. The approach of TAXI [44] for taxonomy induction relies entirely on Hearst-patterns and a string-based system. They used general and domain specific corpora such as GigaWord, ukWac etc. and the common crawl to extract lexico-syntactic patterns. Additionally, they applied pruning methods to refine the generated taxonomy. From the table it can be observed that our system outperforms the NUIG system, which is significant since it is the system which applies comparable techniques to solve this problem. Overall, it seem that for the time being pattern-based systems still outperform the ones founded on distributional semantics. This might, however, change over time as has happened in several other areas of natural language processing.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a semi-supervised heuristic for hyponym-hypernym identification and taxonomy induction from domain specific vocabulary. It identifies significant hyponym-hypernym relations based on a large general corpus, using distributional semantics and a simple string-matching system. We performed an evaluation on by creating taxonomies from several domains and our system shows significant performance for hyponym-hypernym identification and taxonomy induction compared to the current best system based on distributional semantics.

In future version of our system, we plan a closer interaction between the string-based system and perhaps even train the neural network using the output of an initial string-based analysis. Further, we could improve results by including more terms in our word2vec model and adding heuristics to still select a vector for words which are not in the model. Finally, we could blend in well working ideas form other systems like using domain specific corpora and lexical resources to improve the system performance.

ACKNOWLEDGMENT

Bushra Zafar would like to thank computer software department of the NUST university for the expert technical assistance

Michael Cochez performed parts of this research at the Industrial Ontologies Group of the University of Jyväskylä (Finland) and at the Insight Centre for Data Analytics in Galway, Ireland.

Furthermore, it has to be mentioned that the implementation of the software was greatly simplified by Google’s Guava library, Deeplearning4j, and JUnit.

REFERENCES

- [1] M. Geffet and I. Dagan, “The distributional inclusion hypotheses and lexical entailment,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 107–114.
- [2] S. M. Harabagiu, S. J. Maiorano, and M. A. Pasca, “Open-domain textual question answering techniques,” *Natural Language Engineering*, vol. 9, no. 03, pp. 231–267, 2003.
- [3] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.
- [4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Free-base: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250.
- [5] A. Pease, I. Niles, and J. Li, “The suggested upper merged ontology: A large ontology for the semantic web and its applications,” in *Working notes of the AAI-2002 workshop on ontologies and the semantic web*, vol. 28, 2002.
- [6] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [7] D. B. Lenat, “Cyc: A large-scale investment in knowledge infrastructure,” *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.
- [8] E. Lefever, “Lt3: a multi-modular approach to automatic taxonomy construction,” in *9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 2015, pp. 944–948.
- [9] H. Yang and J. Callan, “A metric-based framework for automatic taxonomy induction,” 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*. Association for Computational Linguistics, 2009, pp. 271–279.
- [10] Z. Kozareva and E. Hovy, “A semi-supervised method to learn and construct taxonomies using the web,” in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 1110–1118.
- [11] Z. Kozareva, E. Riloff, and E. H. Hovy, “Semantic class learning from the web with hyponym pattern linkage graphs,” in *ACL*, vol. 8, 2008, pp. 1048–1056.
- [12] M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora,” in *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1992, pp. 539–545.
- [13] G. S. Mann, “Fine-grained proper noun ontologies for question answering,” in *Proceedings of the 2002 workshop on Building and using semantic networks-Volume 11*. Association for Computational Linguistics, 2002, pp. 1–7.
- [14] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Unsupervised named-entity extraction from the web: An experimental study,” *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.
- [15] K. S. Jones, “Experiments in relevance weighting of search terms,” *Information Processing & Management*, vol. 15, no. 3, pp. 133–144, 1979.
- [16] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, “Class-based n-gram models of natural language,” *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.
- [17] D. Lin, “Automatic retrieval and clustering of similar words,” in *Proceedings of the 17th international conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1998, pp. 768–774.

TABLE II
RESULT SUMMARY OF THE PROPOSED SYSTEM AGAINST THE GOLD STANDARD TAXONOMY

Measure	Domain						Average
	Environment (Eurovoc)	Food (WordNet)	Food	Science (Eurovoc)	Science (WordNet)	Science	
Precision	0.2184	0.2042	0.1810	0.2828	0.3515	0.2524	0.2484
Recall	0.2644	0.2746	0.2092	0.3307	0.3152	0.3355	0.2883
F-score	0.2392	0.2342	0.1941	0.3049	0.3335	0.2881	0.2657
F&M	0.1868	0.0036	0.3038	0.1856	0.2165	0.5235	0.2366

TABLE III
COMPARISON OF THE PROPOSED SYSTEM WITH OTHER SYSTEMS

Measure	Baseline	USAAR	TAXI	JUNLP	QASSIT	NUIG	Proposed
Avg. Precision	0.5676	0.5677	0.3318	0.1474	0.1858	0.2200	0.2484
Avg. Recall	0.2369	0.1916	0.3174	0.3009	0.263	0.1100	0.2883
Avg. F-score	0.3330	0.259	0.3212	0.1961	0.2176	0.2028	0.2657
Avg. F&M	0.0046	0.0013	0.2908	0.1498	0.4064	0.0410	0.2366

- [18] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of english words," in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1993, pp. 183–190.
- [19] H. Yang and J. Callan, "Learning the distance metric in a personal ontology," in *Proceedings of the 2nd international workshop on Ontologies and information systems for the semantic web*. ACM, 2008, pp. 17–24.
- [20] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," *Advances in Neural Information Processing Systems* 17, 2004.
- [21] P. Velardi, S. Faralli, and R. Navigli, "Ontolearn reloaded: A graph-based algorithm for taxonomy induction," *Computational Linguistics*, vol. 39, no. 3, pp. 665–707, 2013.
- [22] P. Treeratpituk, M. Khabsa, and C. L. Giles, "Graph-based approach to automatic taxonomy generation (grabtax)," *arXiv preprint arXiv:1307.1718*, 2013.
- [23] R. Navigli, P. Velardi, and S. Faralli, "A graph-based algorithm for inducing lexical taxonomies from scratch," in *IJCAI*, vol. 11, 2011, pp. 1872–1877.
- [24] 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*, 2005, pp. 76–84.
- [25] 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*. Association for Computational Linguistics, 2012, pp. 75–79.
- [26] E. Santus, A. Lenci, Q. Lu, and S. S. Im Walde, "Chasing hypernyms in vector spaces with entropy," in *EACL*, 2014, pp. 38–42.
- [27] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive science*, vol. 34, no. 8, pp. 1388–1429, 2010.
- [28] T. Mikolov and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, 2013.
- [29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, vol. 14, 2014, pp. 1532–43.
- [30] O. Levy, Y. Goldberg, and I. Ramat-Gan, "Linguistic regularities in sparse and explicit word representations," in *CoNLL*, 2014, pp. 171–180.
- [31] N. Shazeer, R. Doherty, C. Evans, and C. Waterson, "Swivel: Improving embeddings by noticing what's missing," *arXiv preprint arXiv:1602.02215*, 2016.
- [32] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Learning hierarchical category structure in deep neural networks," in *Proceedings of the 35th annual meeting of the Cognitive Science Society*, 2013, pp. 1271–1276.
- [33] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu, "Learning semantic hierarchies via word embeddings," in *ACL (1)*, 2014, pp. 1199–1209.
- [34] L. Tan, N. Ordan *et al.*, "Usaar-chronos: Crawling the web for temporal annotations," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 846–850.
- [35] J. Pocostales, "Nuig-unlp at semeval-2016 task 13: A simple word embedding-based approach for taxonomy extraction," *Proceedings of SemEval*, pp. 1298–1302, 2016.
- [36] M. Rei and T. Briscoe, "Looking for hyponyms in vector space," in *CoNLL*. Citeseer, 2014, pp. 68–77.
- [37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [38] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.
- [39] X. Rong, "word2vec parameter learning explained," *arXiv preprint arXiv:1411.2738*, 2014.
- [40] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *HLT-NAACL*, vol. 13, 2013, pp. 746–751.
- [41] P. Maitra, D. Das, and I. Kolkata, "Junlp at semeval-2016 task 13: A language independent approach for hypernym identification," *Proceedings of SemEval*, pp. 1310–1314, 2016.
- [42] 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*, pp. 1315–1319, 2016.
- [43] L. Tan, F. Bond, and J. van Genabith, "Usaar at semeval-2016 task 13: Hyponym endocentricity," *Proceedings of SemEval*, pp. 1303–1309, 2016.
- [44] 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*, pp. 1320–1327, 2016.