Hyperbolic Embeddings for Taxonomy and Facet Extraction

Nathan Bartley
University of Southern California
nbartley@usc.edu

Abstract

Taxonomies are of great importance to knowledge-rich domains like science and to applications like information retrieval. Given the glut of information available online, it is valuable to be able to quickly sort individual documents by their broader topic areas (e.g., their discipline) and their more granular semantics (e.g., their methodology). Hyperbolic embeddings, which incorporate hierarchical information in the data they represent, can be useful for building such taxonomies. In this paper we extend existing taxonomy extraction techniques with hyperbolic embeddings.

1 Introduction

With the massive digitization effort of scientific articles over the last 15 years, we now have access to huge amounts of information, especially in disciplines related to computer science.[1] To take advantage of such a glut of information, researchers must be able to sift through papers to see if they are applicable or interesting. To be able to search through documents and identify the key *concepts* and *facets* that are discussed would be of great value. For example, it is difficult to instantly understand the *methods* being proposed in a paper, the *domain* the authors are working in, nor the *metrics* used to assess their methods.

Similarly, it would be of great interest to examine on a macroscopic level the direction research is taking by such aggregated concepts and facets. How does the adoption of a particular method spread over time?

Siddiqui et al. (2016) posed the problem of extracting such concepts and facets as the Facet Extraction problem: to extract facets is to label each document in a corpus with a ranked list of concepts for each facet. This means that in a paper about computer vision we may have a number of deep neural network models and preprocessing techniques (the *concepts*) be associated with a *technique* facet for that document.

However, the Facet Extraction problem as it stands does not allow for much hierarchy in the concepts nor the facets. If we reframe the problem as one where we jointly label each facet with a list of concepts *and* induce a taxonomy over those concepts, then we can treat the problem as one of taxonomy construction. This may recover better concepts and gain macroscopic insight as to what the corpus is concerned about. Here we describe an extension of an existing taxonomy extraction framework HiExpan to incorporate more hierarchical information through the use of hyperbolic embeddings.

2 Related Work

2.1 Facet Extraction

An early work in facet extraction is Gupta and Manning (2011a), where they characterize a scientific article in terms of its *focus*, *technique*, and *domain*. A focus of an article is its main contribution. A technique is any method or tool used in the article. The domain is the article's application domain. To further illustrate this with an example, an article that concentrates on regularization in RNNs for speech recognition will have a focus of regularization, techniques of regularization and RNNs, and a domain of speech recognition.

To identify the concepts associated with each of the three facets, the authors match a document's text to semantic patterns built on dependency parse trees. Given a set of seed patterns (e.g, a focus pattern is [$present \rightarrow direct\_object$]), the authors bootstrap more patterns from the corpus. After

[1] In the last three years alone there have been nearly 90,000 pre-print articles published on arxiv.org under computer science.

re-weighting the discovered patterns, they identify significant facets in each document. They also topic model their corpus, and tie together the topics with the concepts to analyze the influence different communities have on one another.

Something that is interesting about this work is that they utilize semantic patterns and dependency parse trees to directly extract the relations. However, this work does not consider a richer set of facets for analyzing their corpora. Likewise, they do not rely on hierarchical information in the concepts to inform the influence score.

Another important paper in Facet Extraction is by Siddiqui et al. (2016). In this paper they explicitly define the Facet Extraction problem, and present their framework for extracting concepts and assigning them to arbitrary facets (which can be user-specified). The authors treat the assignment of concepts to facets as a joint optimization over four constructed subgraphs: one with links between concept mentions and topical concepts, one with co-occurrence between concept mentions and section names (e.g., Introduction, Methods, Conclusion), concept mentions and relation phrases, and one with concept mentions and suffix phrases (e.g., "-able" and "-ition"). They then solve this joint optimization problem as a mixed integer programming problem.

This work does not assume a fixed set of facets, and takes advantage of both local sentence-level and corpus-level statistics for multiple levels of granularity in facet extraction across different domains. However, it is computationally expensive, and implicitly models granularity which would be explicitly captured in directly extracting a taxonomy of concepts and facets.

2.2 Taxonomy Construction

Automatic taxonomy construction has been a problem in computational linguistics for many years, as it has been readily apparent the value in automatically organizing a corpus into a well-structured taxonomy to allow for quick information access (or for instance recommendation of new articles). Early methods rely heavily on pre-defined lexico-syntactic patterns for extraction of straightforward "is-a" relations (Hearst, 1992), which gives high precision but very low recall given its fixed patterns.

More recently, work has been done at combining insights from neural language models (namely, using word embeddings trained under the Skip-gram model) and an adaptive recursive hierarchical clustering scheme to construct *topic taxonomies* (Zhang et al., 2018). These topic taxonomies are trees that have many semantically coherent concepts assigned to each node, where the concepts are more granular and specific the further down the tree is traversed.

This work is limited in that it requires a fixed number of clusters for its adaptive clustering module. Relaxing this would allow for a more reliable data-driven taxonomy generation. Similarly, this work is implicitly relying on extracting hypernymy "is-a" relations which limits the possible domain applications of the taxonomies.

Current research in hyperbolic embeddings as applied to concept hierarchies is also constrained by hypernymy relations, as described in Le et al. (2019). In this work, the authors learn a Lorentz hyperbolic embedding model over the Hearst graph (i.e., the graph of "is-a" relations) and evaluate performance by computing for any pair of words the degree to which one is a hypernym of another.

Such pattern-based constraints are addressed by HiExpan (Shen et al., 2018). In this work the authors present a framework that takes a domain-specific corpus, and a task-specific seed taxonomy. With this the authors extract new terms from the corpus, and using an iterative process of set expansion and relation expansion fill out the seed taxonomy. This is all carried out as a joint optimization problem that assigns each term to its appropriate parent node in the taxonomy.

3 Method

In this section we describe the framework used to incorporate dependency parse tree information into the expansion of a seed facet taxonomy.

3.1 Dependency Parse Tree Extraction

In order to increase the extraction of specific mentions of concepts we can associate to facets, we follow a similar approach to Gupta and Manning (2011b). First, we use the SpaCy (Honnibal and Montani, 2017) dependency parser to generate the set of dependency parse trees for some corpus D. Depending on the corpus, we specify a set of trigger patterns.

We also construct the facet extraction pipeline as described in Gupta and Manning (2011b) as a baseline.

3.2 Iterative Tree Expansion

We use the same HiExpan framework as presented in Shen et al. (2018) for iteratively expanding the facet tree laterally and vertically. The input for the framework includes two parts: (1) the corpus D; (2) a seed taxonomy T^0 that is specified by the user. Given the user-specified task, the framework expands T^0 into an expanded taxonomy T. Each node $u \in T$ is a term extracted from D, and each edge (u, v) denotes a task-specific relation. In facet extraction, these represent ”is-a” relations.

The framework works by first extracting phrases from the corpus D, followed by a part-of-speech filter to ensure we recover nouns. After recovering candidate terms, we iteratively expand the width of the taxonomy and the depth. To expand the width of the taxonomy tree we compute the similarities between entities' embeddings (learned via a skipgram model), rank those similarities, and add entities to the set of nodes (in this case, the level of the tree) if the mean reciprocal rank meets a threshold.

To deepen the tree we compute the similarities for the embeddings of the entities to the target parent node and treat the top three most similar entities as candidate children nodes.

When the same node is found in multiple positions in the tree, we treat this as a conflict. We compute the joint similarity for each node and its parents and children to measure the confidence of an entity at each particular node.

3.3 Hyperbolic Embeddings

In order to incorporate the syntactic hierarchical information in the dependency parse trees, we make use of hyperbolic embeddings. Hyperbolic spaces are non-Euclidean spaces with a geometry defined by constant negative curvature. These spaces have received considerable attention recently as they are very useful for hierarchical data. For example, consider some tree t. We can model the exponential branching factor of t in as few as two dimensions: nodes that are l levels deep in t sit on a hyperbolic sphere with radius $r \propto l$, whereas nodes that are fewer than l levels deep sit within the sphere. This is because in hyperbolic geometry disc area and circle length grow exponentially with their respective radius.

There are multiple different equivalent models of hyperbolic geometry: the Klein model, the Poincare disk model, the Poincare half-plane model, and the Lorentz model.

We use hyperbolic embeddings trained on the dependency parse trees for the corpus D. We minimize the following loss function:

$$L(\Theta) = \Sigma_{(u,v)\in D} \log \frac{e^{-d(u,v)}}{\Sigma_{v' \in N(u)} e^{-d(u,v')}} \quad (1)$$

In order to minimize the loss function we follow the following learning procedure:

$$\Theta^{'} \leftarrow \arg\min_{\Theta} L(\Theta) \quad (2)$$

$$\text{s.t. } \forall \theta_i \in \Theta : \|\theta_i\| < 1. \quad (3)$$

where $d(u,v) = \text{arcosh}(1 + 2\frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)})$ (4)

we update Θ by: (5)

$$\theta_{t+1} \leftarrow \text{proj}(\theta_t - \eta_t \frac{(1-\|\theta_t\|^2)^2}{4} \triangledown_E) \quad (6)$$

$$\text{where } \triangledown_E = \frac{\delta L(\theta)}{\delta d(\theta, x)} \frac{\delta d(\theta, x)}{\delta \theta} \quad (7)$$

We use these embeddings in computing the similarities between entities in the facet taxonomies. We use the implementation described in Nickel and Kiela (2018) which makes use of the Lorentz model to learn the embeddings. As there are multiple different models that fulfill the hyperbolic geometry axioms, we can convert between the Lorentz model, which is well-suited to Riemannian optimization, and the Poincare half-disk model which is well-suited and intuitive for visualization, as seen in the visualizations of the dependency parses of the corpora in Fig.1 and Fig.2.

4 Experiments

To test the efficacy of the dependency parse trees in expanding facet taxonomies, we set two research questions:

Q1 Do hyperbolic embeddings of dependency trees increase performance of HiExpan?

Q2 Do the hyperbolic embeddings of dependency trees help construct facet trees?

To answer Q1, we compare the performance of HiExpan on the same DBLP dataset used to test it in (Shen et al., 2018). We use the same tested

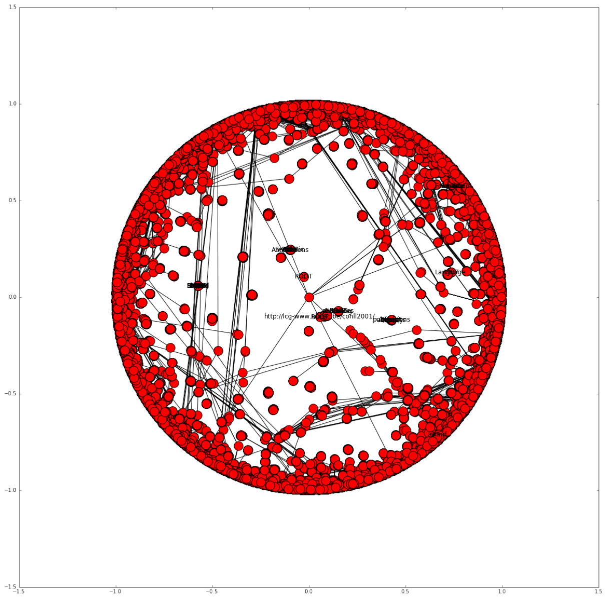

Figure 1: 2 dimensional embedding of the dependency trees on the ACL abstract corpus. Nodes labeled are hubs computed by the HITS algorithm.

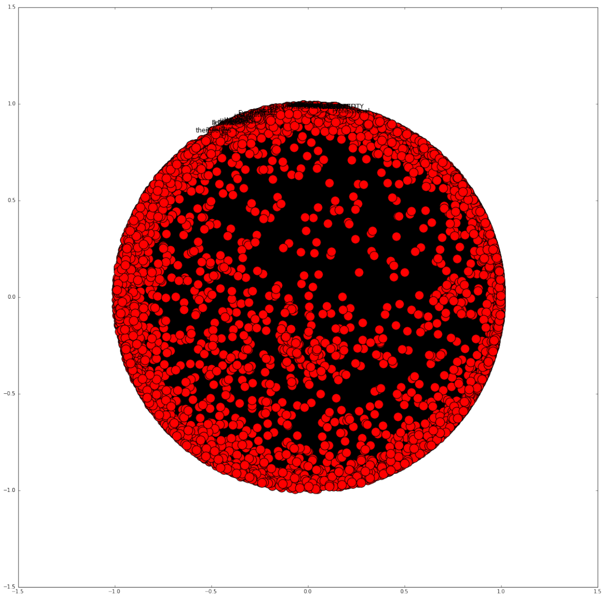

Figure 2: 2 dimensional embedding of the dependency trees on the DBLP abstract corpus. Nodes labeled are hubs computed by the HITS algorithm.

parameters for the basic HiExpan, and compare the model against the model with the hyperbolic embeddings.

To answer Q2, we use the extended HiExpan model and compare its performance against the model from (Gupta and Manning, 2011b) as the baseline model.

4.1 Q1

To test whether the hyperbolic embeddings of dependency trees increase the performance of HiExpan on task-specific taxonomy construction, we evaluate two different versions of HiExpan: one with the original skipgram embeddings, and one with hyperbolic embeddings. We evaluate the two models on the DBLP dataset used in (Shen et al., 2018) which contains 156 thousand computer science abstracts. We evaluate the performance of the two models with **ancestor-F1** and **edge-F1**.

Ancestor-F1 measures whether or not we correctly predict ancestral relations in an taxonomy. We compute it by comparing our predicted taxonomy to a labeled gold-standard in the following way:

$$P_a = \frac{|\text{is-ancestor}_{\text{pred}} \cap \text{is-ancestor}_{\text{gold}}|}{|\text{is-ancestor}_{\text{pred}}|} \tag{8}$$

$$R_a = \frac{|\text{is-ancestor}_{\text{pred}} \cap \text{is-ancestor}_{\text{gold}}|}{|\text{is-ancestor}_{\text{gold}}|} \tag{9}$$

$$F1_a = \frac{2P_a R_a}{P_a + R_a} \tag{10}$$

Similarly, **edge-F1** measures whether or not we correctly predict the edges themselves in a taxonomy (regardless of order). We compute it in the following way:

$$P_e = \frac{|\text{is-edge}_{\text{pred}} \cap \text{is-edge}_{\text{gold}}|}{|\text{is-edge}_{\text{pred}}|} \tag{11}$$

$$R_e = \frac{|\text{is-edge}_{\text{pred}} \cap \text{is-edge}_{\text{gold}}|}{|\text{is-edge}_{\text{gold}}|} \tag{12}$$

$$F1_e = \frac{2P_e R_e}{P_e + R_e} \tag{13}$$

4.1.1 Results

We present the results in Table 1. We compare the results of hyperbolic embeddings on dependency trees against the best configuration of HiExpan as reported in (Shen et al., 2018). We compare the three following hyperbolic embeddings configurations:

- **Min** For every phrase extracted via AutoPhrase, we take the embedding of the word with the minimum L2 norm.

Embeddings	P_a	R_a	$F1_a$
Euclidean (paper)	0.843	0.376	0.520
Euclidean (Impl.)	**0.177**	**0.325**	**0.229**
Hyperbolic (mean)	0.089	0.1706	0.1173
Hyperbolic (min)	0.073	0.135	0.095
Hyperbolic (concat)	0.091	0.091	0.091

Table 1: Experimental results with ancestor-precision, recall, and f1. For the original HiExpan results, paper results are reported as well as the results of our own runs. The bolded results are the best of our runs, which are significantly less than the results of the original paper.

- **Mean** For every phrase extracted, we take the mean over all the constituent word vectors.

- **Concatenate** For every phrase extracted, we sum over all the constituent word vectors.

Results suggest that the original HiExpan with the embeddings over all the terms perform better than the hyperbolic embeddings trained on the dependency parse trees of each sentence in the corpus. We do not present the results for **edge-f1** because they do not change the outcome.

4.2 Q2

To test whether the hyperbolic embeddings of dependency trees help construct facet trees specifically, we compare the model from (Gupta and Manning, 2011b) to two methods:

- a similar bootstrapping approach where we use hyperbolic embeddings to expand the trigger patterns instead of the iterative process described in the paper

- the HiExpan model with the hyperbolic embeddings

We follow the same approach as in (Gupta and Manning, 2011b) and use two annotators to randomly label 30 abstracts from the ACL corpus (22 thousand articles), and compute **edge-f1** for each abstract.

A problem arose in that running the baseline model requires labeled abstracts which were not provided by the original authors. As such, we are currently labelling 30 abstracts to validate the baseline and expanded models.

5 Discussion

The results seem to suggest that the hyperbolic embeddings trained on the dependency parse trees for the DBLP corpus do not help the HiExpan model. Since we were unable to replicate the results of the original paper, we compared the results to our own runs of the model. With this caveat, the original model outperforms the one with hyperbolic embeddings. There are a number of different reasons why this might be the case:

1. The coverage of the extracted terms we were able to get hyperbolic embeddings for was 13,600 terms out of a total of 17,100 extracted from the corpus. The missing terms may have been significant in downstream analysis.

2. We did not restrict the kinds of dependency relations we considered in learning the hyperbolic embeddings. Future work will look into only consider noun-phrase relations like *amod*, *nmod*, *compound*, *obj*.

Future work would entail directly analyzing the hyperbolic embeddings learned on dependency parse trees in how they help extract facet trees per Gupta and Manning (2011b). Dependency parse trees may also not be that effective for this task, and as such other sources of hierarchical information like a constituency parse tree and existing ontologies like WordNet and ConceptNet.

References

Sonal Gupta and Christopher Manning. 2011a. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1–9, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Sonal Gupta and Christopher Manning. 2011b. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th international joint conference on natural language processing*, pages 1–9.

Marti A Hearst. 1992. 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.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. *arXiv preprint arXiv:1902.00913*.

Maximilian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *arXiv preprint arXiv:1806.03417*.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2180–2189. ACM.

Tarique Siddiqui, Xiang Ren, Aditya Parameswaran, and Jiawei Han. 2016. Facetgist: Collective extraction of document facets in large technical corpora. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 871–880. ACM.

Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709. ACM.