Taxonomy refinement using distributional semantics

Rami Aly
Universität Hamburg

November 20, 2018

As part of the project "Shared Tasks for Natural Language Processing" we propose a method to improve existing domain-specific taxonomies by applying a pipeline of refinement steps. Most notably, we introduce the use of Poincaré embeddings as a solution to identify wrong hypernym relationships within a taxonomy as well as to attach disconnected components to a taxonomy. We applied the proposed method to all available domains of the SemEval-2016 Task 13 for the English language and obtained state-of-the-arts results with significant improvements.

1 Introduction

The task of taxonomy induction aims to create a semantic hierarchy of entities by hyponym-hypernym relations, called *taxonomy*, from text corpora . It is a useful tool for tasks such as content organization and retrieval. Compared to many other domains of natural language processing that make use of more recent techniques, like word embeddings and deep neural networks, the task of taxonomy learning is still highly reliable on traditional approaches like extraction of lexico-syntactic patterns (Hearst, 1992) or co-ocurrence information (Grefenstette, 2015). The use of distributional semantics (Mikolov et al., 2013) for hypernym identification has received increasing attraction, however, multiple concerns were raised. Levy et al. (2015) observe that instead of the relation between the two terms x and y, these methods actually learn the independent attribute of a term to be a so called *prototypical hypernym*. Therefore, although usable for hypernym detection, past applications of distributional semantics appear to be rather unsuitable to be directly applied to taxonomies.

We address that issue by introducing a pipeline of refinement steps that employ distributional semantics, in order to improve an existing domain-specific taxonomy. The refinement method deals with the most prominent errors of a taxonomy: Wrong relationships between terms as well as completely disconnected elements. In addition to traditional word2vec embeddings we employ domain-specific Poincaré embeddings. We observe that Poincaré embeddings highly outperform traditional word2vec embeddings. We hypothesize that the difference is explained by the incorporation of the hierarchy between terms within Poincaré embeddings. We apply the refinement pipeline to taxonomies that are generated by the taxonomy induction system TAXI (Panchenko et al., 2016), which achieved the highest score for the above mentioned shared task and uses a combination of lexico-syntactic patterns and substring matching techniques. Therefore, this refinement technique employs distributional semantics as a complementary method on a taxonomy induction system, that is based on more traditional methods. We designed this refinement method to be applied non-restrictively to any domain and language by solely depending on raw text processing. The techniques are applied to all domains of the shared task for the English language.

The reimplementation of TAXI was done by Alex Ossa and me together. However, we worked mainly independently on each topic to improve the existing taxonomy. While Alex employed a deep neural network method, called *Hypenet* (Shwartz et al., 2016) and is going to document his part in a separate report, this paper aims to cover the topic of distributional semantics that I dealt with.

2 Related Work

A taxonomy is generally speaking the classification of entities, things or concepts in a semantic hierarchy. A taxonomy predominantly aims to create a system with relationships, that can be categorized into so-called predications and universally quantified conditionals. While the latter focuses on types, e.g *monkey is a mammal*, predication relationships focus on concrete entities, for example *Einstein is a scientist* (Brachman, 1983). Both relationship types are so called *is-a* relations, in contrast to relations of a meronomy, which organizes entities in parts of a whole by using *has-a* relations. The two entities of a is-a relationship are called hypernym and hyponym, with the first term describing the generic entity and the latter one a specific instance of the entity. Therefore, is-a relationships are asymmetric and transitive, which is why taxonomies are normally hierarchically structured. Phrases that are hyponyms of the same hypernyms are called co-hyponyms, for example *cat* and *dog*, as both could belong to the hypernym *animal* and neither cat nor dog is a hypernym of one another. The shared task focuses on the taxonomies that are created for nouns. Taxonomies based on verb phrases can be created as well, although this is done only sporadically in literature. Creating taxonomies is highly beneficial in numerous NLP tasks, such as personalized recommendations Zhang et al. (2014) or question answering (Yang et al., 2017). The typical workflow of a taxonomy construction technique as described in Wang et al. (2017) consists of two parts. Firstly, the extraction of is-a relationships using either pattern-based or more recently distributional methods and secondly, the construction of the taxonomy from is-a relations using incremental learning, clustering or graph-based induction. Finally, a taxonomy cleansing step is described to remove wrong is-a relations by removing cycles and handling entity ambiguity.

The first shared task on taxonomy extraction was conducted as part of the Semeval 2015 Task 17 (Bordea et al., 2015). The task is concerned with the automatic extraction of taxonomies from text, also called taxonomy induction. The creation of taxonomies based on human compiled resources such as Wikipedia (Mahdisoltani et al., 2013; Suchanek et al., 2007) tend to score better since it is difficult to extract knowledge exclusively from text (Wang et al., 2017). However, these large taxonomies often lack domain-specific and long-tailed knowledge. The task of taxonomy induction addresses this issue, since the use of free-text allows to create domain-specific taxonomies. Thus, the task of taxonomy induction is receiving increasing interest. Since this task is far from being solved, the conducted shared task aimed to contribute to the research of this topic. The organizers provided the participants with respective terms for each domain, on which the taxonomy should be built upon, because the subtask of *Term extraction* is a relatively well-known task and in order to simplify the evaluation. Therefore, submitted systems take the domain name and the respective terms as an input to induce a domain-specific taxonomy. Six different taxonomy creation systems were submitted. INRIASAC and LT3 reached the first and second positions respectively. The latter system used a web corpus constructed by applying BootCat with the provided domain-specific terms as seed terms. They then extracted relations using lexico-syntactic patterns, morphological structure of compound terms and WordNet lookup. INRIASAC also used lexico-syntactic patterns in addition to substring matching and co-occurrence information to extract relationships of the input domain terms on the basis of a corpus from Wikipedia.

The second run of this shared task (Bordea et al., 2016) was extended to a multilingual setting, covering English, French, Italian and Dutch. Moreover, participants were provided with a Wikipedia-based text corpus which was allowed to be manually extended by participants. Five different systems were submitted that year. The second best system, QASSIT (Cleuziou and Moreno, 2016) uses a semi-supervised approach and genetic algorithms. They first calculate the similarity between concepts using semantic vectors and then define a pretopology space from which a taxonomy structure is defined. Finally a genetic algorithm is applied to optimize two parameters: the quality of the added relationships in the taxonomy and the quality of its structure.

The best result for SemEval-2016 Task 13 was achieved by a taxonomy induction system called TAXI, that harvests hypernyms with substring inclusion and Hearst-style lexico-syntactic patterns from domain-specific texts, obtained via language model based focused crawling (Panchenko et al., 2016). Their doctrine is that taxonomy induction should be driven solely on the basis of raw text processing so that a taxonomy can be introduced in a new domain or a new language for which pre-annotated resourced do not exist. Later, we use the extracted taxonomies generated by their method as a baseline system which is then improved by the refinement pipeline. Most submissions of the shared task used traditional pattern-based approaches. An exception is the submission by Pocostales (2016), who applied distributional semantics to construct taxonomies. However, their system scored the lowest out of all submissions (Bordea et al., 2016).

More refined taxonomy induction systems that apply distributional semantics have been introduced since the second shared task was hold. Most recently, Zhang et al. (2018) represent each term as a conceptual topic and defines it as a cluster of semantically coherent concept terms. The topic taxonomy is constructed by using term embeddings and hierarchical clustering, recursively for underlying terms. They further include an adaptive spherical clustering module and a local embedding module to maintain the quality of the recursive process. An expressed limitation is that the number of clusters is a hyperparameter that needs to be pre-specified for the recursive process - thus the number of children for each parent is statically defined.

The task of taxonomy creation offers many unsolved challenges. Wang et al. (2017) recommend to study the combination of pattern-based and distributional methods and how they reinforce each other. Besides Shwartz et al. (2016), there have been very few approaches that use deep learning paradigms, since it is difficult to design a single objective for neural networks to optimize the task. Finally, the task is also insufficiently studied regarding taxonomies for specific domains and under-resourced languages (Wang et al., 2017). Domain knowledge is essential for relation extraction but difficult to obtain. The taxonomy creation should focus on constructing a taxonomy based on knowledge of a specific domain, as it is especially difficult to develop a "one-size-fits-all" taxonomy. Furthermore, domain taxonomies have a higher coverage than existing methods (Alfarone and Davis, 2015).

Our refinement method addresses some of the above mentioned issues by complementary applying a method using domain-specific distributional semantics to a text induced taxonomy, that is based on traditional pattern-based methods.

3 TAXI

The first step of our work consisted in reimplementing TAXI and evaluating some key quantitative statistics. Their method uses a combination of lexico-syntactic pattern and substring matching evaluation on large-scaled domain-specific texts. They take the domain-specific input terms that are provided by the shared task and apply four steps in order to create the taxonomy.

The first source of candidate hypernyms is attained by substring matching. In this method, a substring score is calculated for each input term $\sigma(t_i, t_j) = \frac{length(t_j)}{length(t_i)}$ with t_i being the hypernym term and tj the hyponym term. If we find the same compound word in reverse order, the substring score will be set to zero. If $\sigma(t_i, t_j)$ is larger than three, the relation will be added as a candidate.

The next source of hypernyms is the use of lexico-syntactic patterns. For this purpose, general corpora and domain-specific corpora are crawled based on the input terms. In order to achieve this, general corpora of Wikipedia, CommonCrawl and about 59.2GB of data extracted from English Wikipedia, Gigaword (Parker et al., 2009) ukWac (Ferraresi et al., 2008) and a news corpora from the Leipzig Collection (Goldhahn et al., 2012) are collected. The domain-specific corpora consist of web pages that have been selected by using a combination of BootCat and focused crawling (Remus and Biemann, 2016). Bootcat collects the URLs, that return after sending predefined terms to a Web search engine. These URLs are then used as the seed for a focused crawling method to create a larger corpus of domain-specific web pages. Then multiple systems are applied(PattaMaika (Kluegl et al., 2016), PatternSim (Panchenko et al., 2012), and WebISA (Seitner et al., 2016)), which all use lexico-syntactic patterns, such as the ones described by Hearst (1992). Patterns like *NP*$_0$ *, including NP (... and/or NP)*, with NP$_0$ being the hypernym and NP the hyponym are used to extract noisy is-a relations from the general and domain-specific corpora. Since the extracted hypernyms based on lexico-syntactic contain many noisy relations, an asymmetric pattern-based hypernymy score $\pi(t_i, t_j)$ was introduced. This score combines the information from different hypernym collections and is computed by $\pi^k(t_i, t_j) = \frac{freq^k(t_o,t_j)}{max_j freq^k(t_i,t_j)}$ for the k-th extractor and with $freq^k(t_i, t_j)$ being the frequency of relations between two terms.

The penultimate step consists of combining the scores of hypernym candidates, extracted by the two aforementioned methods. The work proposes two approaches for this step. The first one is a supervised approach, which trains a classifier on the trial dataset provided with the task. It uses two features, namely $\sigma(t_i, t_j)$ and $\pi(t_i, t_j)$. The second method is a static selection of is-a relations. It keeps all hypernyms extracted by substrings $(\sigma(t_i, t_j) > 0)$ and up to two hypernyms extracted by patterns. Although the original paper uses the supervised approach for the English language, we decided to apply the latter approach for our work. This has several reasons; first of all, using the supervised approach requires

System	Domain	R	P	F1	F&M
Original TAXI	Environment	0.2682	0.3382	0.2992	0.2384
Reimplementation	Environment	0.2184	0.3098	0.2562	0.2489
Original TAXI	Science	0.3484	0.3876	0.3669	0.3634
Reimplementation	Science	0.3269	0.4164	0.3663	0.3939
Original TAXI	Food	0.2376	0.3372	0.2787	0.2021
Reimplementation	Food	0.2187	0.3656	0.2736	0.1936

Table 1: Result comparison between original TAXI and our reimplementation baseline.

training data which is not necessarily available for every domain and language (TAXI uses the supervised approach exclusively for English since trial data was only provided for this language). It thus contradicts the initial idea of creating a taxonomy solely on the basis of raw text processing. Secondly, we had difficulties achieving results comparable to the ones in the paper using the supervised approach, although the important parts of their method are openly accessible[1].

The last step consisted of pruning and cleaning the noisy taxonomy. This step needed to be re-implemented by us. To identify cycles in the taxonomy, we employed the Tarjan algorithm (Tarjan, 1972). If a cycle is detected, one relation will be removed at random. However, in practice at most two relationships were removed for all domains. That is why we decided, that this simplistic approach is sufficient for most cases. Moreover, we attached all unconnected components to the root (this does not include single disconnected nodes). Lastly, we remove all relationships that have the domain root as a direct or indirect hyponym. Thus, we make sure that the domain root is also the root of the created taxonomy. This step is now part of the official TAXI github, after we made a pull request to include that implementation.

Altogether, the displayed results for the baseline method differ slightly from the original paper for the food and science domain and quite substantially for the environment domain, with our results being worse. The results of the reimplementation and the original TAXI results are listed in Table 1 for all three domains (environment, science, food) of the shared task.

We further evaluated the contribution of the two methods(substring matching and lexico-syntactic patterns) to the total score for the science domain by only executing one of the methods at a time and by comparing the scores. The method using substring matching achieved a score of 0.268 while the pattern matching approach scored 0.11. Therefore, both approaches significantly improve the score although substring matching appears to be the more effective method. We further notice that both methods complement each other very well, as the score using both approaches is very similar to the sum of using them separately.

4 Taxonomy Refinement Method

The proposed method uses the existing taxonomy created by the TAXI system, its corresponding domain terms and raw-text corpora to create an improved version of the input taxonomy. The structure of the taxonomy is not restricted, it can be connected or consists

[1] https://github.com/tudarmstadt-lt/taxi

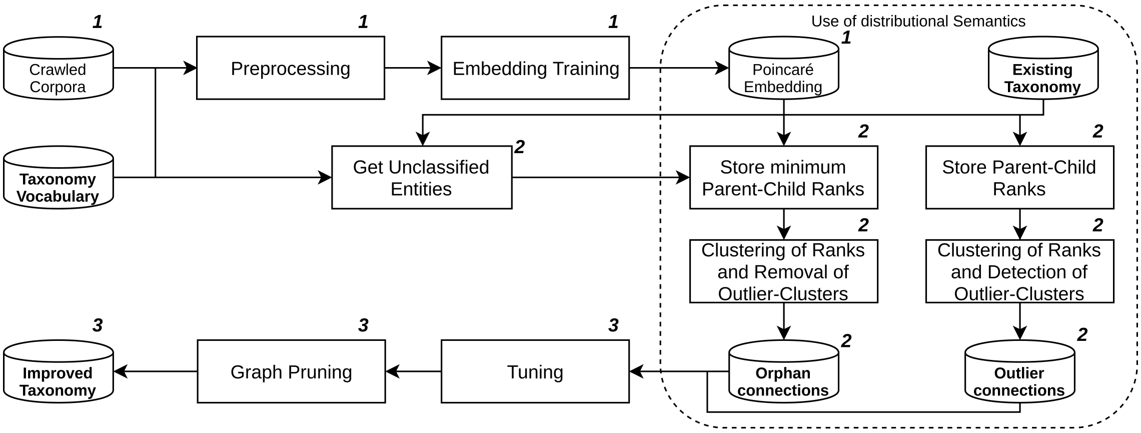


Figure 1: Outline of the underlying process to improve an existing taxonomy. The numbers correspond to the step within the pipeline.

of cycles and completely disconnected domain terms. We apply a pipeline of three steps, two of which directly impact the existing taxonomy. The main focus of our refinement method is the use of distributional semantics, in particular Poincaré embeddings. By using them we aim to deal with the most prominent errors of a taxonomy: wrong relationships between terms as well as completely disconnected elements. The first step consists of creating domain-specific Poincaré embeddings (Section 4.1). The trained embeddings are then used to identify wrong hypernym relationships in the taxonomy as well as to attach unconnected terms to the taxonomy (Section 4.2). In the last step, we further optimize the taxonomy, remove cycles and attach remaining disconnected terms to the root (Section 4.3).

4.1 Domain-specific Poincaré Embedding

We aim to construct Poincaré embeddings that are designed for the specific domain of the taxonomy, since using task-specific embeddings have been shown to be more effective than general-purpose embeddings (Li et al., 2016; Wang et al., 2017). To create suitable Poincaré embeddings we use noisy hypernym relationships, extracted from a combination of general and domain-specific corpora, comparable to the is-a relationships extracted by TAXI. The extracted domain-specific relationships are further filtered to create more accurate but still noisy relationships.

Cleaning For the next step, the extracted noisy relationships of the common and domain-specific corpora are further processed separately and combined afterwards. Let V be the set of all domain terms of a given taxonomy. In order to limit the amount of words and relationships the Poincaré embeddings have to learn, we decide to limit the is-a relationships R on pairs for which both entities are part of the taxonomy's vocabulary. By using only vocabulary of the domain-specific taxonomy, the thus created collection of relationships is domain-specific as well. Furthermore, every relationship is removed, which is not above a fixed frequency threshold T_f. We further remove every reflexive relationship, so that $\forall a \in V : (a, a) \notin R$ holds true. We could include symmetric relationships, however since R is relatively noisy, we decided on only keeping the more frequent pair so that $\forall (a, b) \in R : (f(a, b) - f(b, a)) > T_f$ is fulfilled, with f being a function that maps the a pair to the number of occurrences in the collection of extracted relationships. Hence, R

is transformed to being antisymmetric and irreflexive. Same procedure is applied to all relationships extracted by the common-crawl corpora which are then added to R. However, a is-a relation is only added if it does not contradict the properties of R, resulting in a collection of relationships in which relationships gathered by common-crawl corpora can only expand but not change R as created by the domain-specific corpora.

Training These cleaned is-a relationships are then used to train Poincaré embeddings. We choose a dimensionality of 50 and 400 epochs to ensure convergence. We trained an additional model on noun pairs extracted from Wordnet[2]. Finally, we also trained word2vec embeddings (Mikolov et al., 2013) on a wikipedia corpus (Table A1), connecting compound words by a character '_' to be able to natively learn distributional semantics for compound words. This appeared to be a more suitable approach for this task, since combining vector representation of subwords has shown to be difficult for combinations such as *signal processing*. The specific hyperparameters for every embedding model are listed in Appendix A1.

4.2 Distributional Semantics

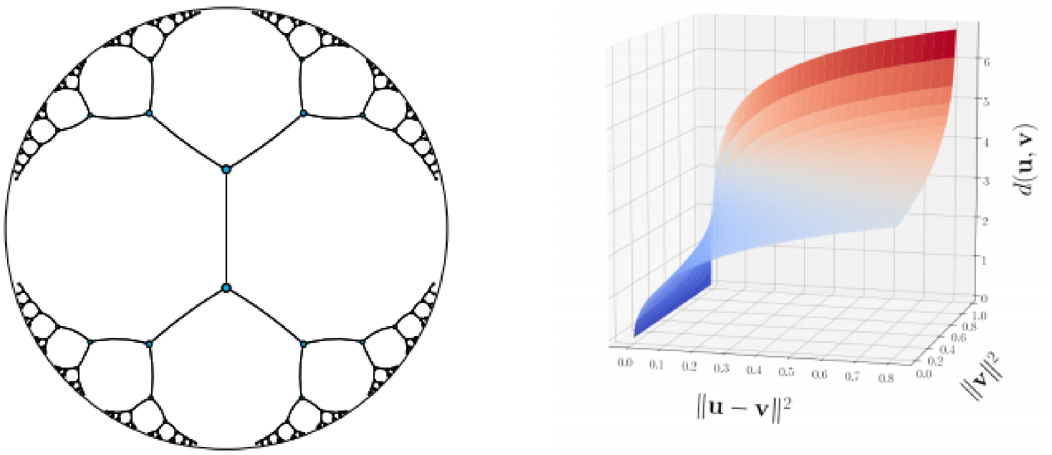
In contrast to word2vec which is computed in euclidean space and commonly applies the cosine distance (similarity) $\frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|}$ as a similarity measure, Poincaré embeddings use a hyperbolic space, specifically the Poincaré Ball. Poincaré embeddings explicitly capture the hierarchy between words in the embedding space and aim to capture hierarchical relationships more efficiently. The distance between two points $\mathbf{u}, \mathbf{v} \in \mathcal{B}^d$ for a d-dimensional Poincaré Ball model is defined as:

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

The Poincaré distance as defined in Eq. 1 enables us to capture the hierarchy and similarity between words simultaneously. The Poincaré distance increases exponentially with the depth of the hierarchy. So while the distance of a leaf node to most other nodes in the hierarchy is very high, nodes on lower level such as the root, have a comparably small distance to all nodes in the hierarchy, as seen in Figure 2c). Thus, a small distance between two nodes is achieved by a combination of similarity between terms and their respective position in the hierarchy. Naturally, not all terms of a taxonomy are represented in the trained embeddings. During this step we ignored hypernyms for which no distributional representation exists.

Outlier Identification Firstly, we use distributional semantics to detect wrong hypernym relationships in the taxonomy. The Poincaré embeddings are used to compute a rank between every child and parent of the existing taxonomy. The rank(x,y) is defined as the index of y in the list of all sorted Poincaré distances of all entities to x. In order to detect wrong hypernyms, called *outliers*, we employ k-means on all calculated ranks. The cluster with the highest ranked hypernym is marked. If the cluster is additionally not the center cluster (cluster with most hypernyms) we mark it as an outlier cluster and corresponding relationships are accordingly removed from the taxonomy. This step is repeated until no

[2] https://wordnet.princeton.edu/



(b) Embedding of a tree in $\mathcal{B}^2$

(c) Growth of Poincaré distance

Figure 2: Visualization of Poincaré embeddings, in b) the length of every edge is of equal length in $\mathcal{B}^2$. c) visualizes the Poincaré distance d(u,v) in relation to the norm $||v||^2$ and the direct distance to another node. Illustration taken from Nickel and Kiela (2017)

.

outlier cluster remains. The system that uses word2vec aims to find co-hyponym relationships, as word2vec similarity does not capture parent-child relationships. Therefore, we compute the distance to the closest co-hyponym (child of the same parent) for every node. The clustering technique as just introduced is then applied to identify outliers.

Orphan Insertion This step aims to reattach disconnected nodes, called *orphans*, to the respective taxonomy. This includes orphans that were already disconnected in the input taxonomy, as well as orphans that were produced by the removal of relationships in the previous step. For every term that is not connected to the taxonomy, the parent with the lowest ranked is searched and stored as a potential parent. Similar to aforementioned approach, we apply k-means to the collection of ranks of all potential parents in order to identify the relationships that will be added. All relationships apart from the ones that belong to outlier clusters (if existent) are added to the taxonomy. The word2vec system applies a similar method. However, the identified relationships are not registered as parent-child relationships but as co-hyponym relationships. Thus, a link is added between the parent of the most similar co-hyponym in the taxonomy and the orphan.

4.3 Improved Taxonomy

The final step consists of further improvements of the taxonomy and its structure. In case a distributional representation does not exist for a term, we used a heuristic to determine how to connect an orphan; if a substring of a compound word can be found in the hierarchy, the orphan is linked as a hyponym to that entity. Otherwise, the orphan remains disconnected. This substep depends on the underlying language and possibly needs to be adjusted. All other steps should work independently of language or domain.

Finally, we repeat the final step of the TAXI system, since the refinement method could theoretically create circles and disconnect components. Thus, we connect disconnected components to the root. We further identify circles again and remove them if detected.

5 Evaluation

The proposed methods are evaluated on the SemEval-2016 challenge for Taxonomy Extraction Evaluation Bordea et al. (2016). We evaluate the proposed methods on the English language sub-task. The baseline taxonomy we refine is the reimplementation of the TAXI system as described in Section 3.

Metrics The employed metrics were selected with the goal to exhaustively assess the quality of the taxonomy. Quantitative evaluation is done by comparing them against the gold standard of the corresponding domain. The main metrics we employed are recall, $\frac{|E_S \cap E_G|}{|E_G|}$, precision $\frac{|E_S \cap E_G|}{|E_S|}$, and F_1 score $\frac{2*(P*R)}{(P+R)}$. Additionally, we employed the Fowlkes&Mallows measure defined in (Velardi et al., 2013; Wagner and Wagner, 2007). It measures how well a taxonomy clusters similar nodes by calculating a score $B^i_{S,G} = \frac{n^i_{11}}{\sqrt{(n^i_{11}+n^i_{10}) \cdot (n^i_{11}+n^i_{01})}}$ for every cut i in the taxonomy, with n_{00} being the number of object pairs that are in different clusters in the target taxonomy S as well as the gold standard taxonomy G and in same clusters for n_{11}. n_{10} and n_{01} are pairs that are in the same cluster in the target taxonomy but not in the gold taxonomy and vice versa. The Cumulative Fowlkes&Mallows Measure is defined as $B_{S,G} = \frac{\sum_{i=0}^{k-1} \frac{i+1}{k} B^i_{S,G}}{\frac{k+1}{2}}$. We can see that this score benefits from a node that is connected to the root, even if the relation is wrong as some cuts will receive a higher score. Therefore, this score is additionally a good metric to evaluate the connectivity of a taxonomy.

6 Results and Discussion

Domain	Method	Recall	Precision	F1	F&M
Environment	Baseline	21.84	30.98	25.62	24.89
Environment	Root	22.61	20.07	21.26	**41.58**
Environment	Word2Vec	22.99	28.30	25.37	28.50
Environment	Poincaré Wordnet	22.61	29.35	25.54	27.26
Environment	Poincaré custom	**26.05**	**38.52**	**28.27**	30.08
Science	Baseline	32.69	41.64	36.63	39.39
Science	Root	32.69	29.92	31.24	**55.92**
Science	Word2Vec	36.56	40.77	38.55	48.19
Science	Poincaré Wordnet	36.56	44.50	40.14	44.41
Science	Poincaré custom	**38.28**	**44.72**	**41.25**	45.21
Food	Baseline	21.87	36.56	27.36	19.36
Food	Root	22.18	19.70	20.87	**43.18**
Food	Word2Vec	23.31	33.36	27.45	26.36
Food	Poincaré Wordnet	27.16	**40.02**	32.36	25.35
Food	Poincaré custom	**29.30**	38.85	**33.41**	28.01

Table 2: Comparison between baseline results and refinement methods applied to all three domains of the SemEval2016 Task. Highest result for each domain is emphasized.

A summary of the results of all applied systems on the SemEval 2016 Task 13 task are shown in Table 2. The root method which simply connects all orphans to the root of the taxonomy has the highest F&M score, as every node is connected to the taxonomy. However, it lacks in every other score significantly. The refinement method using word2vec did only increase the score for the science domain. The word2vec embeddings connect more orphans to the taxonomy as seen in Table 4, however, the quality of connections does not appear to be comparable. Both Poincaré embeddings outperform the word2vec embeddings and they improve the baseline taxonomy in all scores, with a bigger increase in recall than in precision. The improvements appear to increase with the size of the taxonomy, as the score for environment domain ($|V| = 261$) increased the least and the the score for food ($|V| = 1556$) the most.

Ablation study For this ablation study we analyze the scores when either outliers are removed or when orphans are connected to the taxonomy. Table 3 shows the respective scores as well as the actual score when both methods are combined. We can see that the food domain mostly benefits from the orphan insertion step, possibly since the food domain has the highest amount of unconnected nodes. However, both algorithms improved the taxonomy almost to an equal degree when applied to the environment domain. We further notice, that the algorithms appears to be complementary to each other as the separate improvements basically sum up, when combined. They are partly even higher than the sum, since identified outliers possibly create new orphans which are considered for the orphan insertion.

Domain	Method	Orphan	Outlier	Final
Environment	Baseline	—	—	25.62
Environment	Poincaré Wordnet	25.49	25.40	25.22
Environment	Poincaré Custom	26.67	26.65	28.21
Science	Baseline	—	—	36.63
Science	Poincaré Wordnet	39.49	37.80	40.58
Science	Poincaré Custom	39.86	37.13	41.69
Food	Baseline	—	—	27.36
Food	Poincaré Wordnet	31.83	27.90	32.18
Food	Poincaré Custom	33.37	27.52	33.47

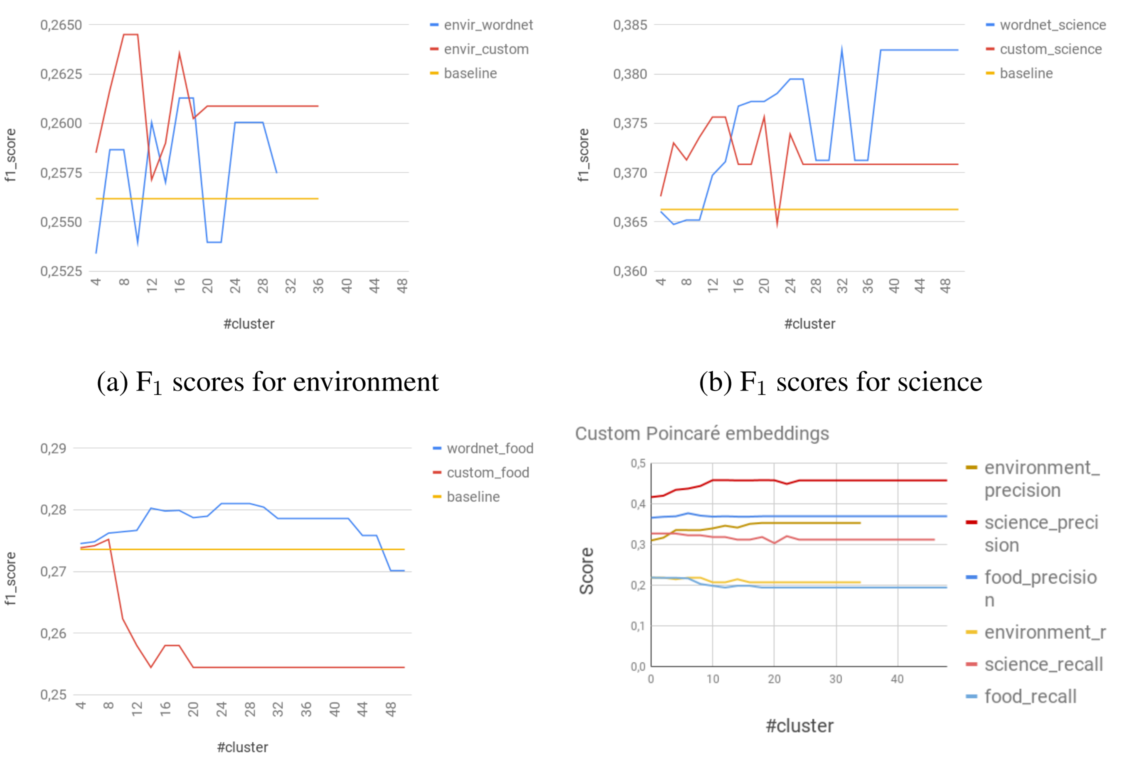
Table 3: F_1 Scores with removed outliers or inserted orphans. Furthermore, the scores of both methods and the baseline for each Domain are shown. The final scores slightly differ from Table 2 as the last two sub-steps of Step 4.3 have not been applied.

Comparison to Wordnet's Poincaré and Word2Vec The results of word2vec embeddings were sub-par. The interpretation of word similarity as co-hyponyms for our refinement method does not appear to be appropriate. However, using word2vec as a means to detect hypernyms has shown to be rather unsuitable since Levy et al. (2015) observed that even more complicated methods such as the *diff* model (Fu et al., 2014; Wang et al., 2017) do not actually learn a relation between two terms - they rather learn the independent attribute of a term to be a so called *prototypical hypernym*. This information however, does not appear to be suitable for a taxonomy refinement method, since the identification of wrong relationships within the taxonomy and the attachment of new nodes needs to take the relationship between two terms of the taxonomy into account. Based on the results we achieved by using Poincaré embeddings, we hypothesize that their attributes result in a system that learns is-a relationships between terms by taking both words into account. The Wordnet embeddings only beat the taxonomy-specific embeddings regarding the precision score for the food domain. In every other aspect the custom embeddings outperform Wordnet by a substantial margin. Poincaré embeddings that are trained on Wordnet's hypernym relations are naturally more accurate. This can be seen in Table 3 for the step which removes outliers. The results of Wordnet are higher on average and appear to not be as much affected by different number of clusters as shown in Figure 3. However, the advantage of the custom embeddings enfold for the orphan insertion step, as seen in Table 3. Most orphans either occur rarely or they are complex compounds like *thermal discharge* and *protection of plant life*. Table 4 shows for each domain and embedding the number of orphans that were recognized and connected to the taxonomy. For two of three domains, the taxonomy-specific embeddings recognize almost two times the amount of terms.

Domain	Word2Vec	Wordnet	Custom
Env.	58	18	35
Science	69	31	33
Food	257	147	257

Table 4: Number of orphans of the baseline taxonomy that were reattached by using different embeddings

K-means as a selection method An important hyperparameter for the refinement method is the number of clusters k for the k-means algorithm. As stated at the beginning, the pipeline was created with the idea in mind to be solely driven on the basis of raw text processing, since it makes the method more flexible and applicable to a variety of domains and languages. We thus avoided the use of supervised approaches, since they would require annotated data. To use a static threshold also appeared unsuitable, since the distance and/or the ranks between entities of a taxonomy shifts with the size and structure of the taxonomy. A simple clustering technique such as k-means seems to be more suitable. The relevance of a selection method is much more important for the outlier removal than for the orphan insertion step. Figure 3 shows the F_1-Scores for both Poincaré embeddings for varying cluster number k as well as the baseline score. With exception of the food domain, the score increases mostly for any number of clusters, although a small k appears to be more effective. As a general rule we observe that a bigger k results in more removed relationships. This is most likely due to the main cluster becoming smaller with greater k because the granularity increases. As seen in Figure 3d, which plots the recall and precision scores with increasing k for the custom Poincaré embeddings, the precision only increases for small k, while the recall further decreases. That observation is reasonable as a smaller k creates less but more confident identifications of outliers, as their rank deviation from the mean is most likely larger.



(a) F_1 scores for environment

(b) F_1 scores for science

(c) F_1 scores for food

(d) Precision and recall scores for custom Poincaré embeddings

Figure 3: Scores of taxonomy with removed outliers for different number of clusters k.

7 Conclusion

This work demonstrated a pipeline for improving an existing taxonomy through the use of distributional semantics, in particular Poincaré embeddings. They showed to be able to identify wrong hypernym relationships and add correct ones to the existing taxonomy so that the overall quality of the taxonomy improves. Poincaré embeddings outperformed the word2vec model significantly in the applied method. We further showed that the use of Poincaré embeddings trained on relationships, that were extracted on basis of a specific domain by raw text processing, are superior to ones trained on Wordnet. This rather simple use of Poincaré embeddings already leads to substantial improvements, so that additional applications and research of Poincaré embeddings is encouraged. Furthermore, we plan to apply the method to other languages to ensure its non-restrictive applicability. Based on the results of this experiment it would be highly interesting to further investigate, whether Poincaré embeddings address the raised concerns regarding word2vec embeddings, as a means for recognizing lexical inference relations.

References

Alfarone, D. and Davis, J. (2015). Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1434–1441, Buenos Aires, Argentina.

Bordea, G., Buitelaar, P., Faralli, S., and Navigli, R. (2015). Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, CO, USA. Association for Computational Linguistics.

Bordea, G., Lefever, E., and Buitelaar, P. (2016). Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, CA, USA.

Brachman, R. J. (1983). What IS-A is and isn't: An analysis of taxonomic links in semanticnetworks. *IEEE Computer*, 16(10):30–36.

Cleuziou, G. and Moreno, J. G. (2016). Qassit at semeval-2016 task 13: On the integration of semantic vectors in pretopological spaces for lexical taxonomy acquisition. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1315–1319, San Diego, CA, USA.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, Marakech, Morocco.

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (2014). Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1199–1209, Baltimore, MD, USA.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey.

Grefenstette, G. (2015). Inriasac: Simple hypernym extraction methods. *arXiv preprint arXiv:1502.01271*.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545, Nantes, France.

Kluegl, P., Toepfer, M., Beck, P.-D., Fette, G., and Puppe, F. (2016). Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, CO, USA.

Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., and Sycara, K. P. (2016). Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In *the 26th International Conference on Computational Linguistics*, Osaka, Japan.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2013). Yago3: A knowledge base from multilingual wikipedias. In *Conference on Innovative Data Systems Research 2013*, Asilomar, CA, USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Stateline, NV, USA.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30*, pages 6338–6347, Long Beach, CA, USA.

Panchenko, A., Faralli, S., Ruppert, E., Remus, S., Naets, H., Fairon, C., Ponzetto, S. P., and Biemann, C. (2016). Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, CA, USA.

Panchenko, A., Morozova, O., and Naets, H. (2012). A semantic similarity measure based on lexico-syntactic patterns. In *Proceedings of KONVENS 2012*, pages 174–178, Vienna, Austria.

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2009). English gigaword forth edition. In *Linguistic Data Consortium*, Philadelphia, PA, USA.

Pocostales, J. (2016). Nuig-unlp at semeval-2016 task 13: A simple word embedding-based approach for taxonomy extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1298–1302, San Diego, CA, USA.

Remus, S. and Biemann, C. (2016). Domain-specific corpus expansion with focused webcrawling. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia.

Seitner, J., Bizer, C., Eckert, K., Faralli, S., und Heiko Paulheim, R. M., and Ponzetto, S. (2016). A large database of hypernymy relations extracted from the web. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, Portorož, Slovenia.

Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2389–2398, Berlin, Germany.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

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

Wagner, S. and Wagner, D. (2007). Comparing clusterings an overview. arXiv:1609.04747.

Wang, C., He, X., and Zhou, A. (2017). A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203, Copenagen, Denmark.

Yang, S., Zou, L., Wang, Z., Yan, J., and Wen, J.-R. (2017). Efficiently answering technical questions - a knowledge graph approach. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 31111–3118, San Francisco, CA, USA.

Zhang, C., Tao, F., Chen, X., Shen, J., Jiang, M., Sadler, B., Vanni, M., and Han, J. (2018). Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*, pages 2701–2709, London, United Kingdom.

Zhang, Y., Ahmed, A., Josifovski, V., and Smola, A. (2014). Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 243–252, New York City, NY, USA.

Appendix

Parameter	Value
Noisy relationships common	27,612,520
Noisy relationships food	11,109
Noisy relationships science	1,514
Noisy relationships environment	863
T_f common	5
T_f domain	3
Cleaned relationships $\|R\|$	3102
Poincaré dim	50
Poincaré epochs	400
Word2vec corpus size	3.1GB
Word2vec dim	300
Word2vec min_count	5
Word2vec epochs	30
$k_{word2vec}$	3, 6 (orphan, outlier)
$k_{wordnet}$	2, 20 (orphan, outlier)
k_{custom}	2, 6 (orphan, outlier)

Table A1: Parameters, settings and attributes of corpora, embeddings and clusters for different domains. The selected k for each embedding is kept the same for all domains of the shared task.