Task guided taxonomy construction

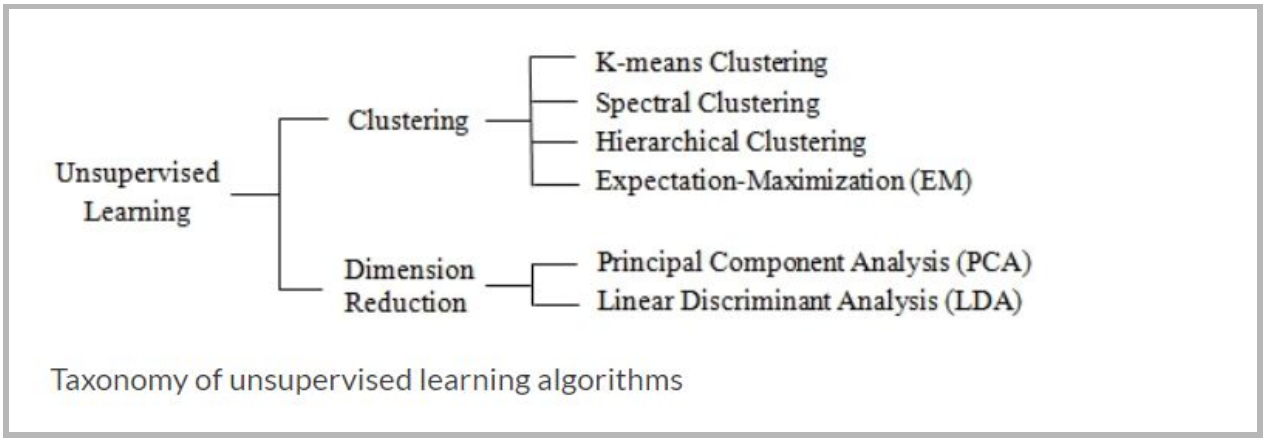
Sanchita Badkas
2019H1030520P

Introduction :

This report talks about the HiExpan algorithm used for a novel approach called Task Guided Taxonomy Generation. The HiExpan is based on the concept of Set expansion and weakly supervised relationship extraction. This paper would first discuss these concepts before diving into the main algorithm. Also, at the end of the discussion the paper suggests some methods for increasing the applicability of the algorithm along with the quality.

Taxonomy Construction:

Taxonomy has been around for a long time for naming and classifying entities in a particular domain. An example of taxonomy for Unsupervised machine learning can be given as



Taxonomy of unsupervised learning algorithms

[25]

Manual taxonomy construction is laborious and subjective.As a result, automatic taxonomy construction has been researched a lot[7][8][9][10]. However, most of these studies assume the availability of a domain specific corpus. [9] gives an example of construction of medical taxonomy is done using corpus of biomedical literature, life science journals and medical books. However,

1. Sometimes domain specific corpus is unavailable
2. As the domains start becoming more and more specific characterizing text corpus becomes difficult.

An approach used for constructing domain specific taxonomies used in[11][12] refers to creation of a bag of words using domain specific keywords and then using clustering methodologies on the available corpus. However, the relationships between those keywords is

ignored in this approach. Thus, keeping track of context is important while doing taxonomy construction.

Other approaches [13][14][15]], leverage the pattern based distribution to extract 'is-A' relationships from the corpus. However, they don't cover

1. Multilevel flexible hierarchies like 'city-state-country'
2. Diversity in taxonomies as they cover domain specific taxonomies limiting their applicability.

Thus, task guided taxonomy was a novel approach proposed in [1] which takes a "seed" taxonomy tree as task guidance along with domain-specific corpus for generating the desired taxonomy.

Relationship extraction:

The task of extracting semantic relationships from a given corpus/document is referred to as Relation Extraction (RE). RE is essential for extracting structured information from unstructured data. For e.g. [23]consider the following text,

'*CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York. '*

The following relationships can be extracted from the above text,

Subject	Relation	Object
American Airlines	subsidiary	AMR
Tim Wagner	employee	American Airlines
United Airlines	subsidiary	UAL

Different approaches for relation extraction are:

1. <u>Rule-based RE</u>

   Extracting relations based on predefined rules. E.g. "*USA is a country*" here _ is a _ would be a rule. Simplicity and ease of extraction of relations for a specific domain are major advantages of this approach. However, due to a huge variety

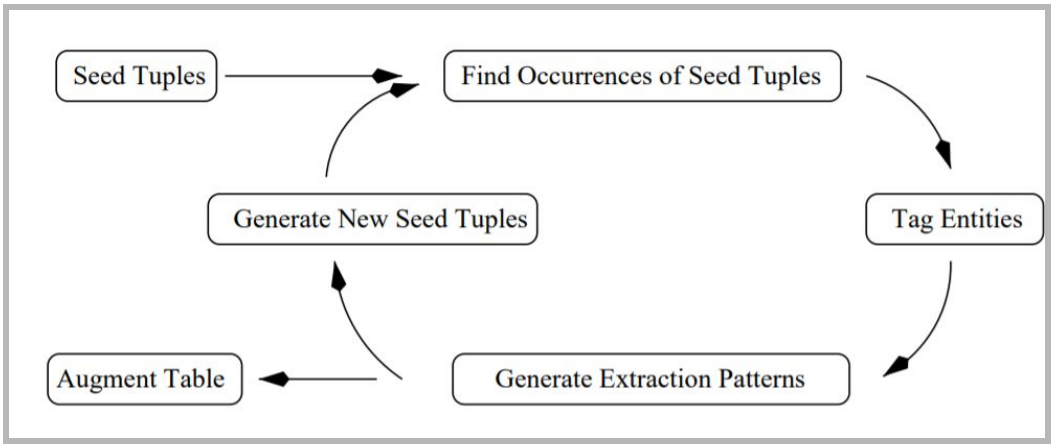
in language a lot of human labour is required for creating rules that would all the relations from a corpus. Thus, scaling this to corpus level is difficult.

2. Supervised RE

   This approach includes use of machine learning models like neural networks, binary classifiers to determine if there is a relation between entities. However, the corpus needs to be processed by some NLP modules in order to extract features like context words, part-of-speech tags, tokens, proximity distance, dependency path between entities. These features are then fed to the models who determine whether the relationship exists or not. The high quality supervision provides us with high recall scores. However, it's expensive to label examples in order to train the data.

3. Weakly supervised RE

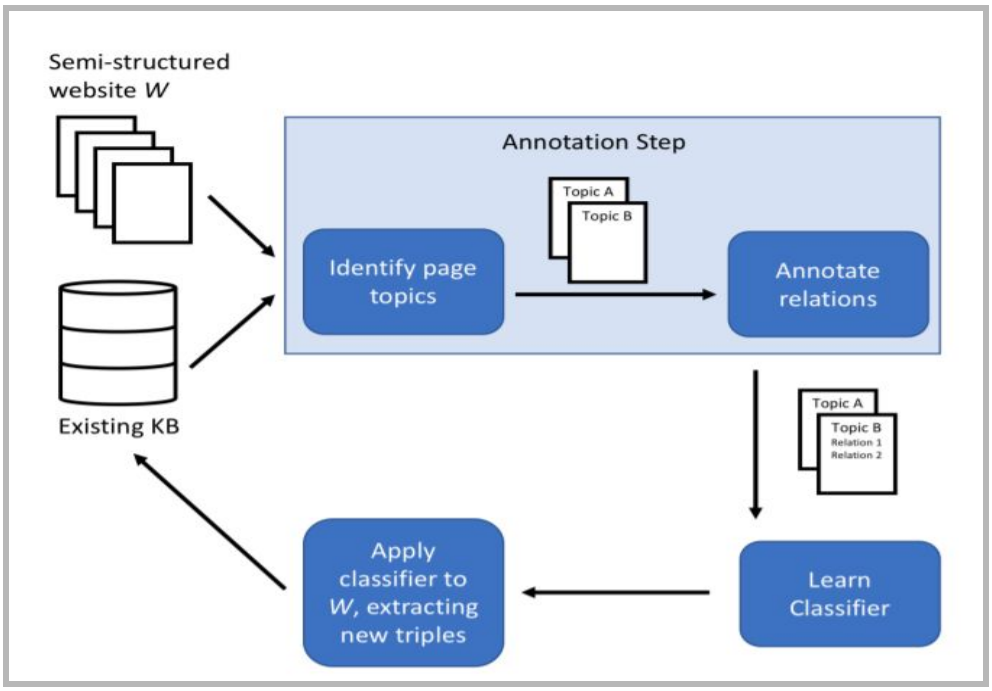
   Recently a lot of methods for RE have involved the use of Neural Networks. These systems are extremely accurate and provide a good amount of coverage for RE on corpus level. However, in order to train the models a huge amount of training data is required, which is not always the case. Thus, this approach is useful when we start out with a handful of seed rules and using an iterative process find new ones. For e.g. *{Hydrogen, Oxygen, Argon, Nitrogen}* are gases which can be fed to the system who will find the rest of the elements from the corpus. SnowBall[16] is one of the examples of this approach which is explained using the below figure.



   One of the major drawbacks of this method is when an error is introduced in the cycle, the model becomes more error prone with each iteration. An example for the same is given in [5]. The paper mentions an example of a seed set of elements *{iron, mercury, carbon}* here mercury is a planet, a Greek god as well as an element. When the error is introduced, all planets and Greek gods would be added to the seed set of elements.

4. Distantly supervised RE

   This approach works by combining the above two approaches. The supervised RE approach is used, however the seed set is taken from existing knowledge bases such as WikiLists, DBpedia, Freebase, Yago etc which is iteratively improved using techniques who use the weakly supervised RE approach . CERES[24] is an example of this type, whose working is shown in the figure.This tremendously reduces the manual labour, but is restricted to the knowledge base. Also, like weakly supervised RE this approach is also a bit prone to iterative errors.



5. Unsupervised RE

   This approach is not entirely unsupervised as it uses some heuristics and general rules in order to extract relations from a corpus. This approach used almost no labelled examples but is completely dependent on the heuristics.

Weakly supervised relationship extraction[3]:

We already discussed the weakly supervised RE approach in brief above. It utilizes a small amount of seed relation instances to extract more such instances of the same relation by consolidating redundant information from the corpus.

There are two types of weakly supervised RE methodologies as follows[22]:

1. Pattern based approach

   This is used to learn the textual pattern for relationship extraction. It predicts the relationship between an entity pair from multiple sentences who include both the entities.

   The traditional patterns extract the patterns in a bootstrapping manner. This method, however, faces difficulties during matching of the learned patterns to unseen context which leads to semantic drift. An e.g for the same is given in [3] as follows:

   Consider two sentences as,

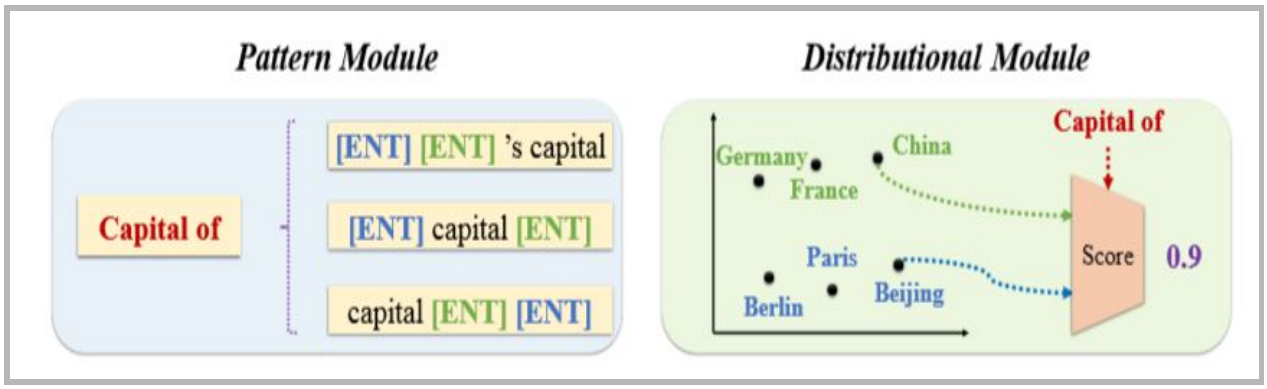
   - *Beijing , the capital of China, is a megacity rich in history.*
   - *Tokyo , Japan's capital , was originally a small village .*

   Here, the pattern X the capital of Y is extracted easily. But, faces difficulty while extracting the same pattern in the second sentence due to the variety in the language.

2. Distributional approach

   These approaches use the corpus level co-occurrence statistics of entities. They focus on learning the low-dimensional representations to preserve these statistics which leads to entities having similar semantic meanings having similar representations.

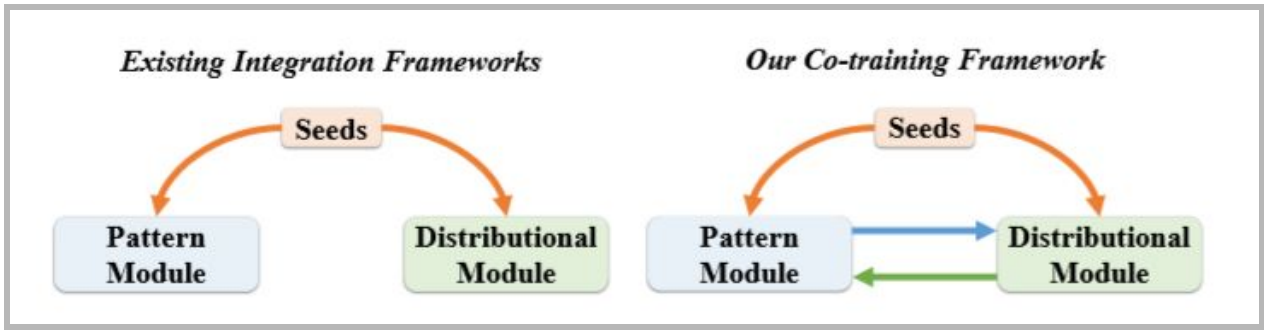
These approaches can be understood clearly with the help of following figure,



REPEL - Relation Extraction with Pattern-enhanced Embedding Learning[3]:

The approaches used for weakly supervised RE extract relations from the corpus using two different perspectives. Thus, the RE results can be enhanced by combining these two methods.

There are many approaches proposed which use this idea[21]. Though all these approaches show good results, a drawback of these models is that the supervision of the frameworks comes completely from the initially provided relation instances. Due to the sparsity of available relation instances, we choose weakly supervised RE, but the above mentioned approaches would perform better only if more training instances are provided.
This forms the premise of [3]. The framework proposed by them is represented by the following figure.



Both the modules are used to provide extra supervision to each other which was lacking in the initial approaches.
The paper treats the pattern module as a generator to extract candidate seed instances from the corpus while the distributional module acts as a discriminator to evaluate each instance. These evaluated instances serve as extra signals to adjust the generator. In turn, the generator will generate highly confident instances which act as seeds to improve the discriminator. This way, extra supervision is provided to both the modules.

We would now briefly discuss the above modules mathematically,

1. Pattern Module

   Patterns are discovered amongst the entities having the shortest dependency path between two entities.

   $R(\pi) = |G(\pi) \cap Spair| / |G(\pi)|$

   Where,

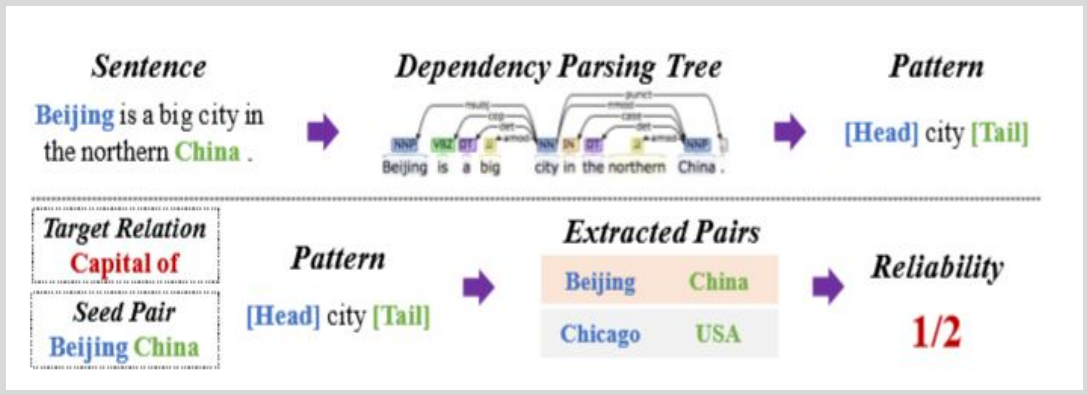
   Π - Represents the patterns

   $R(\pi)$ -Represents the reliability of the p

   $G(\pi)$ - Represents all the entity pairs extracted by the pattern

   S_{pair} - Represents the set of seed entity pairs under the target relation

   Thus, the numerator of $R(\pi)$ is the number of seed entity pairs which can be discovered by the pattern,and the denominator counts all extracted entity pairs. This explained with an example from the paper,



2. Distributional Module

   The paper uses [3] to build a bipartite network between all entities and words[6]. The weight between the entity and a word is defined as the number of sentences in which they co-occur. For an entity 'e' and a word 'w', the conditional probability is inferred as

   $p(w|e) = \exp(x_e \cdot c_w) / Z$

   Where,

   X_e is the vector representation of entity

   C_w is the embedding vector of word w

   Z is the normalization

   Both the functions of the modules are simplified to form objective functions which are optimized.

For module interaction, another objective function is introduced whose goal is to encourage the agreement of both the modules. This objective function is referred to as the joint optimization problem introduced in the paper.

SetExpan:

As discussed in the previous section of weakly supervised relation extraction, a small set of seed is given to the algorithm who extracts similar relationships from the given corpus. This process of expanding the original set of seeds is termed as set expansion. Significant amount of work has been proposed in [18][19][20]. These studies termed as Google Set, SEAL and Lyretail give good quality results. In these approaches, a query consisting of seed entities is submitted to the search engine to mine top ranked web pages. However, this seed oriented online data extraction is costly. Thus the studies of offline processing was focused on. These studies are categorized based on approaches they use,

1. One time entity ranking

2. Iterative pattern based bootstrapping

Let's discuss these approaches in brief,

Previous approaches:

1. <u>One time entity ranking:</u>

   This approach uses the assumption that similar entities would appear in similar contexts. Thus, studies under this approach make a one time ranking of candidate entities based on their distributional similarity with the seed entities. Context required for the algorithms are brought in by wikipedia lists and free text pattern and entity-entity distributional similarity is calculated based on all context features. However, since all the contexts are used, seed intrusion error might take place like the example mentioned in the weakly supervised section of this paper. Also, the extractions take place based on the initial seeds provided.

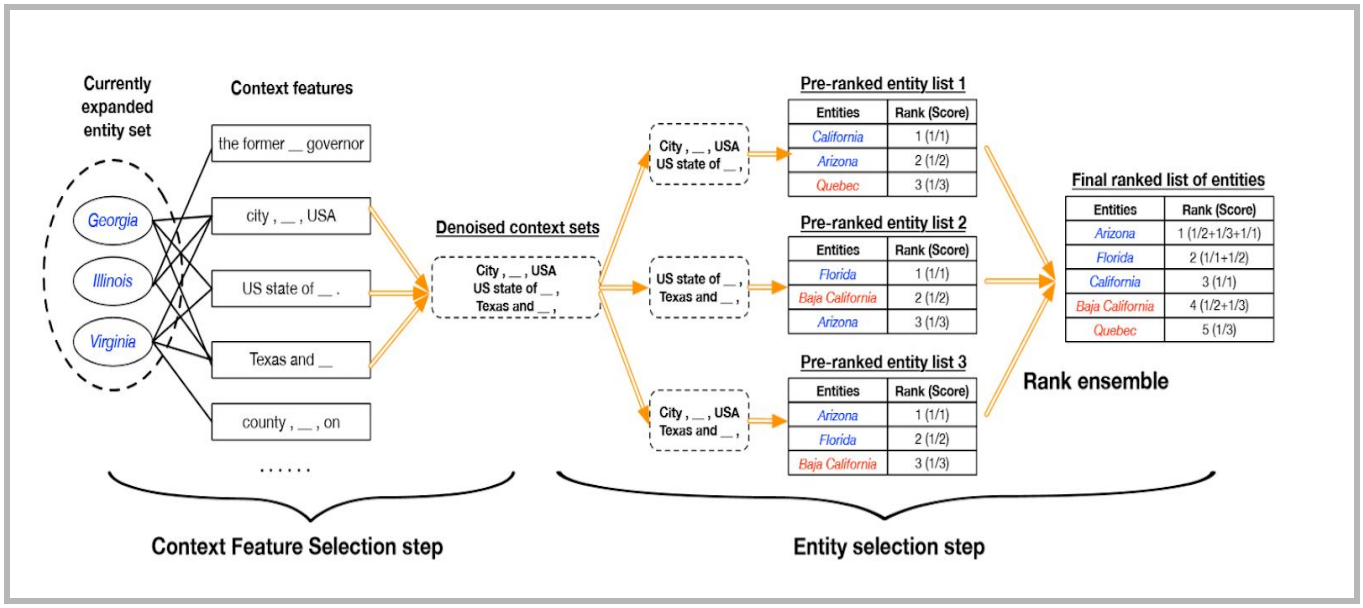
2. <u>Iterative pattern based bootstrapping</u>**:**

   This approach starts from seed entities and extracts quality patterns, based on a predefined pattern scoring mechanism, and it then applies extracted patterns to obtain even higher quality entities using a different entity scoring method. Thus, the process iteratively accumulates high quality patterns which are used for future iterations. However, the quality of these patterns needs to be strictly maintained. The seed intrusion problem can grow exponentially with each iteration causing semantic shift. Thus, the entity scoring methods are crucial and due to sensitivity towards the patterns, a huge amount of caution needs to be exercised. However, having a perfect scoring mechanism is difficult due to diversity in the text data.

The SetExpan paper[2] proposes a new set expansion framework to address these challenges.

1. To overcome the seed intrusion problem, it selects the context features based on distributional similarity instead of using all the available contexts.
2. To overcome the semantic drift problem, they reset the feature pool at the beginning of each iteration. They make use of an unsupervised ranking based ensemble method at each iteration to refine the entities.

At every iteration the following steps take place,



Let's discuss in brief both the above steps,

1. <u>Data model and context features:</u>
    a. Data is modeled as a bipartite graph, with candidate entities on one side and their context features on the other. These features are obtained using skip grams and coarse grained types.
    b. Between each pair of entity 'e' and context feature 'c' weight is calculated using TD-IDF transformation.

$$f_{e,c} = \log(1 + X_{e,c}) \left[\log |E| - \log \left(\sum_{e'} X_{e',c} \right) \right]$$

Here,
$X_{e,c}$ is the raw co-occurrence count between entity e and context feature c
|E| is the total number of candidate entities
In this case, they have treated each entity e as a "document" and each of its context feature c as a "term".

2. <u>Context dependent similarity:</u>
For expanding the entity set, we need to find the set of entities that are most similar to the current set. Weighted Jaccard similarity measure is used for this purpose. Given a set of context features F, context-dependent similarity is calculated as follows:

$$Sim(e_1, e_2|F) = \frac{\sum_{c \in F} \min(f_{e_1,c}, f_{e_2,c})}{\sum_{c \in F} \max(f_{e_1,c}, f_{e_2,c})}.$$

3. Context feature selection:

This step helps find a feature subset F^* of fixed size Q that best "profiles" the target semantic class. Therefore, given such F^*, the entity-entity similarity conditioned on it can best reflect their distributional similarity with regard to the target class. However, this is an NP hard problem. Thus, a heuristic method that first scores each context feature based on its accumulated strength with entities in X and then selects top Q features with maximum scores is used.

4. Entity selection via rank ensemble:

In this step, the algorithm ranks each candidate entity based on its score calculated by the following formula and then adds top-ranked ones into the expanded set.

$$score(e|X, F) = \frac{1}{|X|} \sum_{e' \in X} Sim(e, e'|F).$$

Here, X is the currently expanded set.

However, due to the ambiguity of natural language in free-text corpora, the selected context feature set F may still be noisy in the sense that an irrelevant entity is ranked higher than a relevant one. Thus, a sampling without replacement method to generate T subsets of context features F_t,t =1,2,...,T is used. Each subset is of size α |F|where α is a model parameter within range [0,1]. For each F_t, pre-ranked list of candidate entities L_t based on score(e|X,Ft) is obtained using the previous equation. r_i^t is used to denote the rank of entity e_i in list L_t. If entity ei does not appear in L_t, $r_i^t = \infty$. Finally, T pre-ranked lists and scores each entity based on its mean reciprocal rank (mrr) are collected. All entities with average rank above r, namely mrr(e)≤ T/r, will be added into entity set X.

$$mrr(e_i) = \sum_{t=1}^{T} \frac{1}{r_t^i}, \qquad r_t^i = \sum_{e_j \in E} I\left(score(e_i|X, F_t) \leq score(e_j|X, F_t)\right)$$

where I(·) is the indicator function. Naturally, a relevant entity will rank at top position in multiple pre-ranked lists and thus accumulate a high mrr score, while an irrelevant entity will not consistently appear in multiple lists at high position which leads to low mrr score.

Drawback of these approaches:

SetExpan faces challenges in terms of generating high-quality as follows:

1. Modelling global taxonomy
    a. In case a term appears in multiple expanded sets a conflict resolution policy would be required.
2. Cold start
    a. All these methods have an initial seed set. But what if we wanted to provide an empty seed set. In this case, the generator will need to refer to another same context taxonomy tree for generating the seed set. That is, in case wish to generate the taxonomy for a subdomain we won't be able to do it without providing the seeds of that subdomain.

Task guided taxonomy construction:

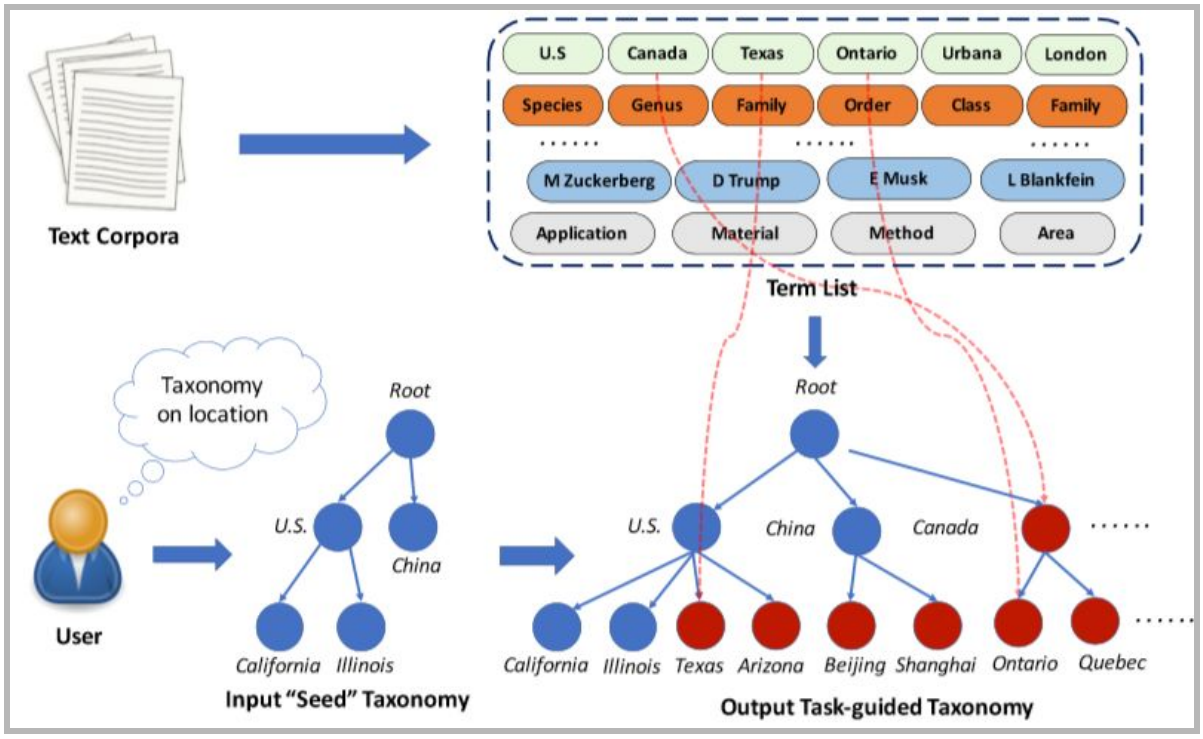
In order to overcome the above mentioned challenges, [1] proposed a new approach called 'Task guided Taxonomy construction.

The algorithm used for this purpose was termed as HiExpan which consists two modules,

1. During the tree expansion process a confidence score is calculated for putting the term in each position and the one with the highest score is chosen. Also, at the end of the hierarchical tree expansion process and global optimization is performed.
2. Weakly supervised relation extraction is performed to infer the parent child relationship and find seeds for a specific parent.

The two methods we saw earlier, REPEL and SetExpan form the based on these two modules who help in recursive expansion of seed sets.

The following figure explains the basic function of this algorithm.



Briefly the algorithm works as follows,

1. Text corpora and a seed taxonomy is provided as input.
2. Hierarchical taxonomy construction takes place in two ways:
    a. Width Expansion -

        This is where the SetExpan algorithm is used. In the above figure, once the seed taxonomy is provided containing two US states, 'California' and 'Illinois' the algorithm expands this taxonomy horizontally by finding other US states like 'Texas' and 'Arizona'.

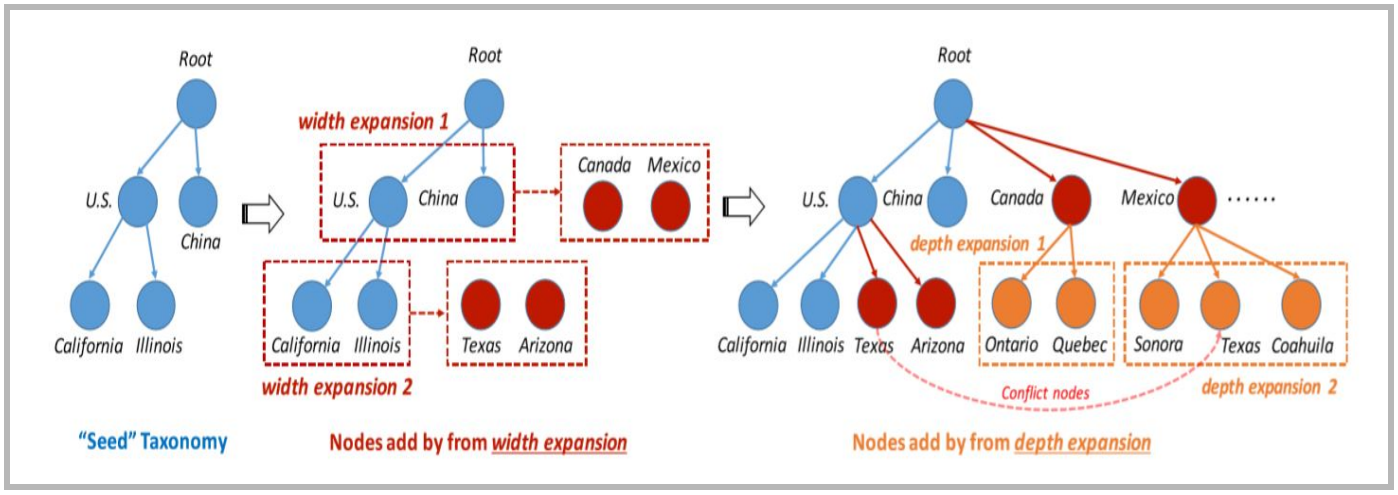
    b. Depth Expansion -

        This is where the REPEL algorithm is used. In the above figure, the Canadian states of 'Ontario' and 'Quebec' would be found only after 'Canada' is found. When we find 'Canada' after the width expansion, we don't have any seed for it's child nodes. This is referred to as cold start. By making use of the found contexts with help of US's seed we expand the 'Canada' node using depth expansion.

    c. Conflict resolution -

        As the provided seed is very weak, care needs to be taken to ensure that the nodes introduced in the first iterations are of high quality as, in case the quality is compromised the further iterations would lead the tree in the wrong direction.

    d. The above processes take place iteratively, until the complete tree is constructed.



The algorithm is given below,

Algorithm 1: Hierarchical Tree Expansion.

Input: A seed taxonomy $\mathcal{T}^0$; a candidate term list V; maximum expansion iteration *max_iter*.
Output: A task-guided taxonomy $\mathcal{T}$.

$\mathcal{T} \leftarrow \mathcal{T}^0$;
for *iter from 1 to max_iter* **do**
  $q \leftarrow queue([\mathcal{T}.rootNode])$;
  while *q is not empty* **do**
    $e_t \leftarrow q.pop()$;
    ⊡ ***Depth Expansion***;
    if $e_t.children$ *is empty* **then**
      $S \leftarrow$ DEPTH-EXPANSION(e_t);
      $e_t.children \leftarrow S$;
      $q.push(S)$;
    ⊡ ***Width Expansion***;
    $C_{new} \leftarrow$ WIDTH-EXPANSION$(e_t.children)$;
    $e_t.children = e_t.children \oplus C_{new}$;
    $q.push(C_{new})$;
  ⊡ ***Conflict Resolution***;
  Identify conflicting nodes in $\mathcal{T}$ and resolve the conflicts;
Return $\mathcal{T}$;

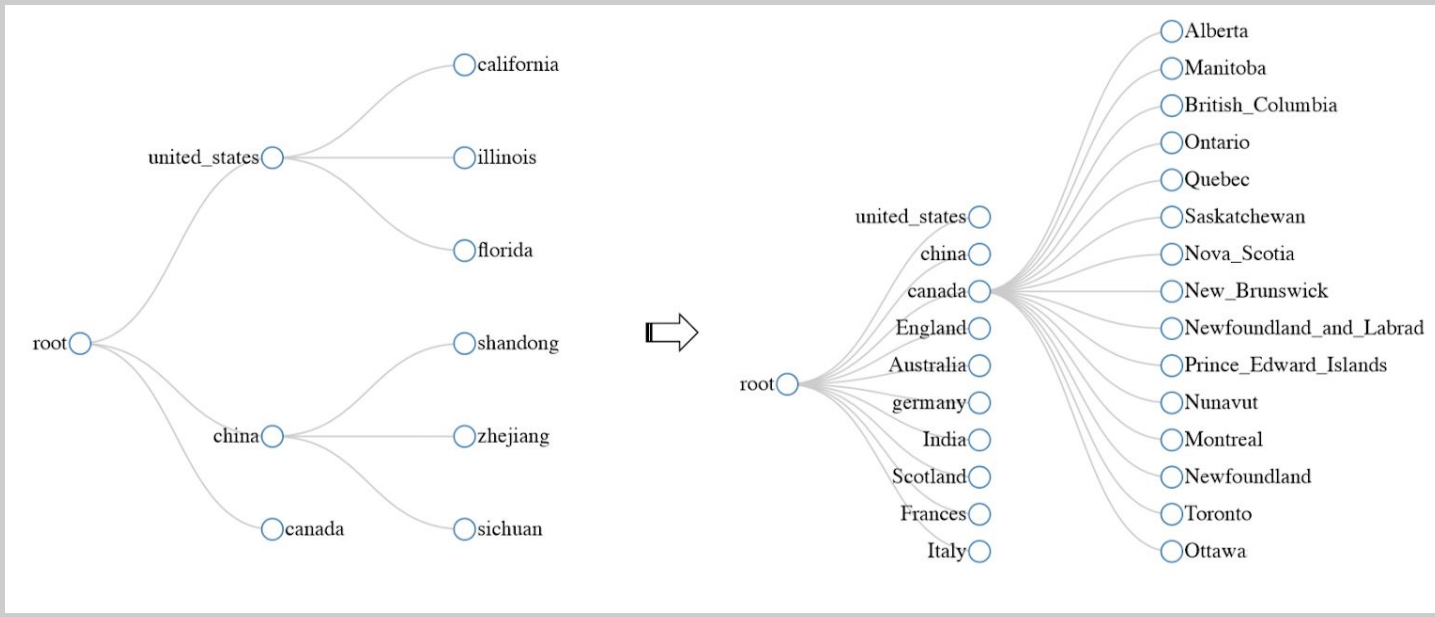
Since this is the first time this kind of algorithm was introduced, there wasn't any available baseline to compare it against.
Thus, the study used variations of the algorithm for comparative analysis.
These include

1. HSetExpan - Iteratively applies SetExpan at each level of taxonomy. For lower levels it uses the child-parent similarity for deciding the best parent to attach the node.
2. NoREPEL - The HiExpan algorithm without the REPEL module. It instead uses a skip gram [17]model for learning term embeddings.
3. NoGTO - HiExpan algorithm without the optimization module.
4. HiExpan - The proposed framework.

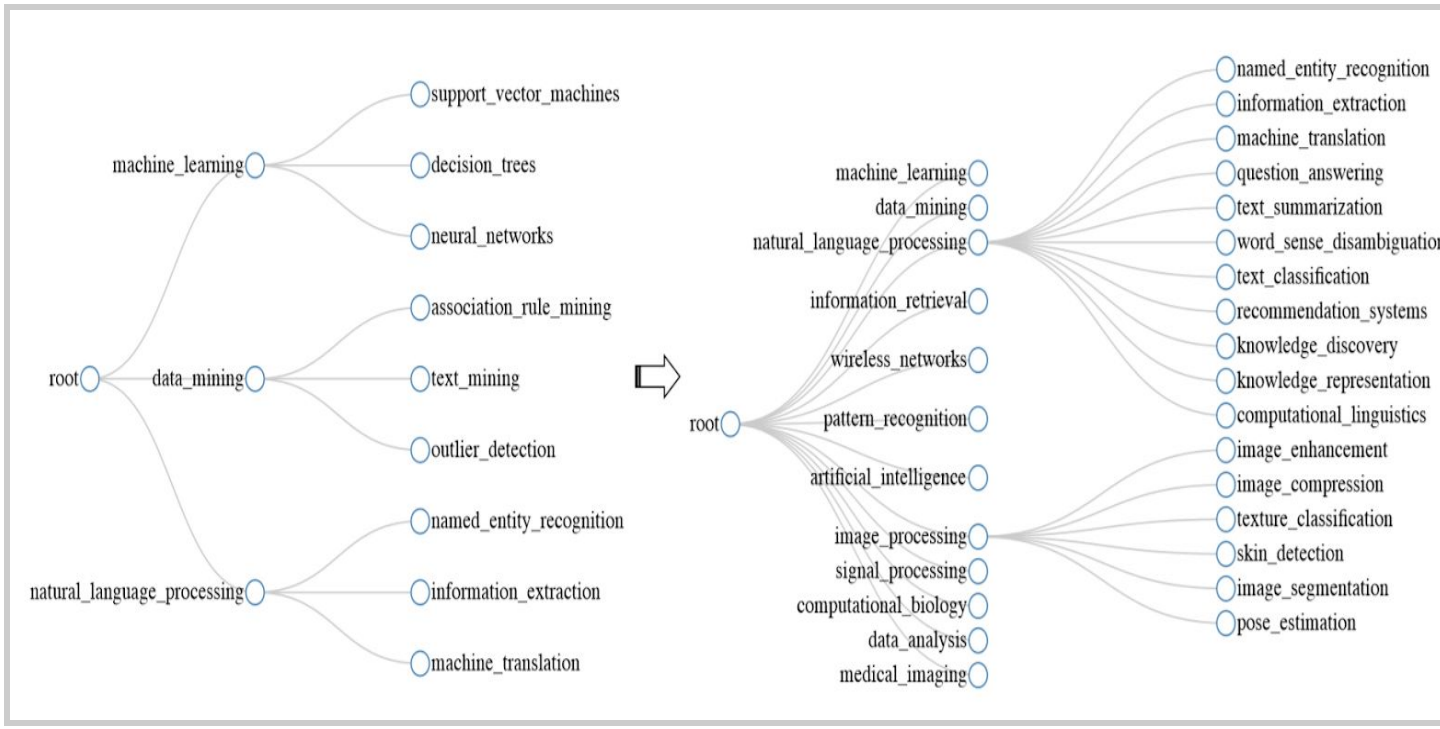
The datasets used for comparative analysis were

Dataset	File Size	# of Sentences	# of Entities
Wiki	1.02GB	1.50M	41.2K
DBLP	520MB	1.10M	17.1K
PubMed-CVD	1.60GB	4.48M	36.1K

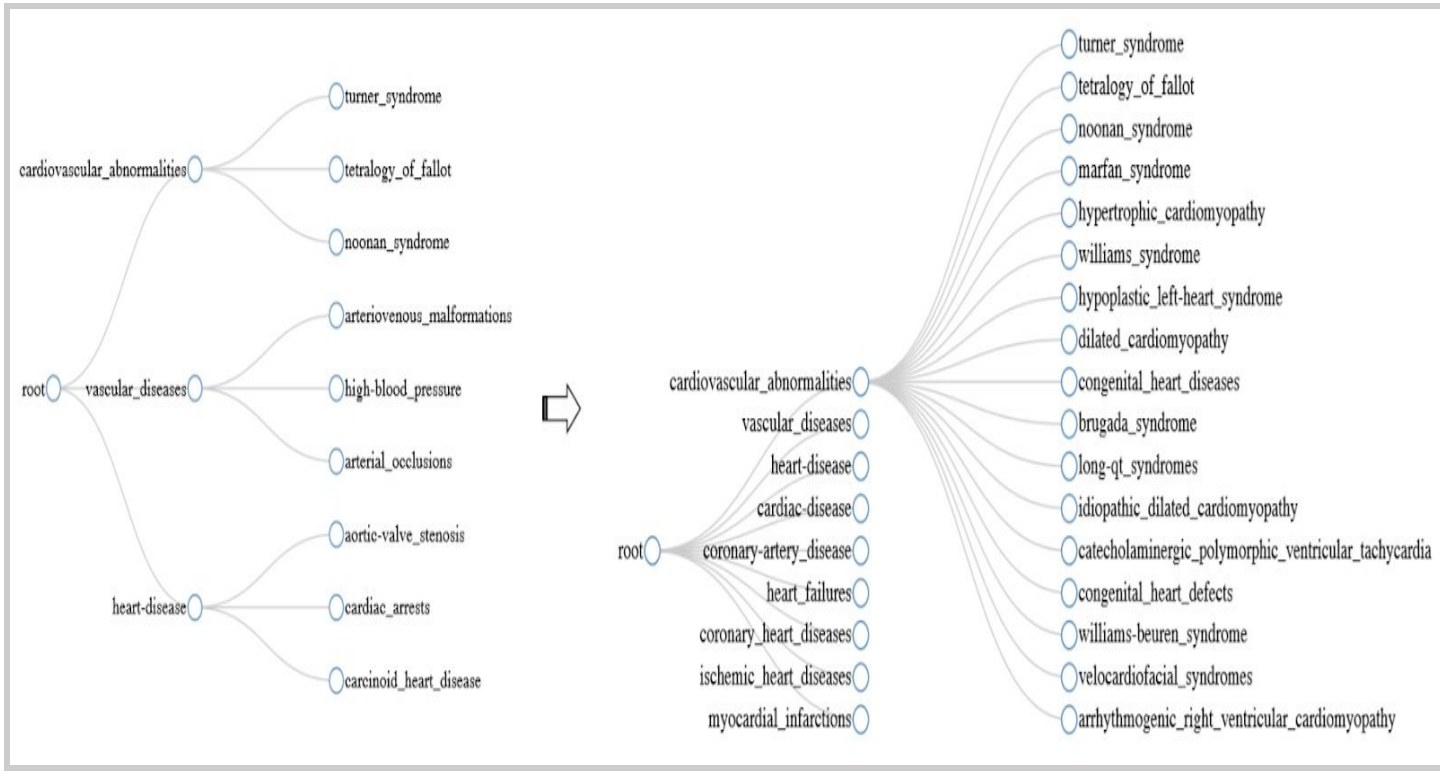
For the Wiki dataset, the country taxonomy was provided as seed as in the figures above. The results were,



For the DBLP dataset, the results were,



For the PubMed dataset,



The quantitative results included P_a, R_a, $F1_a$ denote the ancestor-Precision, ancestor-Recall, and ancestor-F1-score, respectively. Similarly, edge-based metrics are P_e,R_e,and $F1_{e,}$respectively.

The formula for calculating the ancestor values are,

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

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

$$F1_a = \frac{2P_a * R_a}{P_a + R_a},$$

The Ancestor-F1 measures correctly predicted ancestral relations.It enumerates all the pairs on the predicted taxonomy and compares these pairs with those in the gold standard taxonomy. The gold standard taxonomy was constructed using human labour.

The results were,

Method	Wiki						DBLP						PubMed-CVD					
	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$	P_a	R_a	$F1_a$	P_e	R_e	$F1_e$
HSetExpan	0.740	0.444	0.555	0.759	0.471	0.581	0.743	**0.448**	**0.559**	0.739	0.448	0.558	0.524	0.438	0.477	0.513	0.459	0.484
NoREPEL	0.696	0.596	0.642	0.697	0.576	0.631	0.722	0.384	0.502	0.705	**0.464**	0.560	0.583	**0.473**	0.522	0.593	**0.541**	0.566
NoGTO	0.827	0.708	0.763	0.810	0.671	0.734	0.821	0.366	0.506	0.779	0.433	0.556	0.729	0.443	0.551	0.735	0.506	0.599
HiExpan	**0.847**	**0.725**	**0.781**	**0.848**	**0.702**	**0.768**	**0.843**	0.376	0.520	**0.829**	0.460	**0.592**	**0.733**	0.446	**0.555**	**0.744**	0.512	**0.606**

The results show important roles played by the REPEL and optimization algorithms. In the case of DBLP, HiExpan solved conflicts by removing the nodes in the currently expanded taxonomy. As a result in the same amount of iteration HiExpan generated a smaller tree compared to HSetExpan.
Thus, HiExpan was significantly effective.

What can be improved?

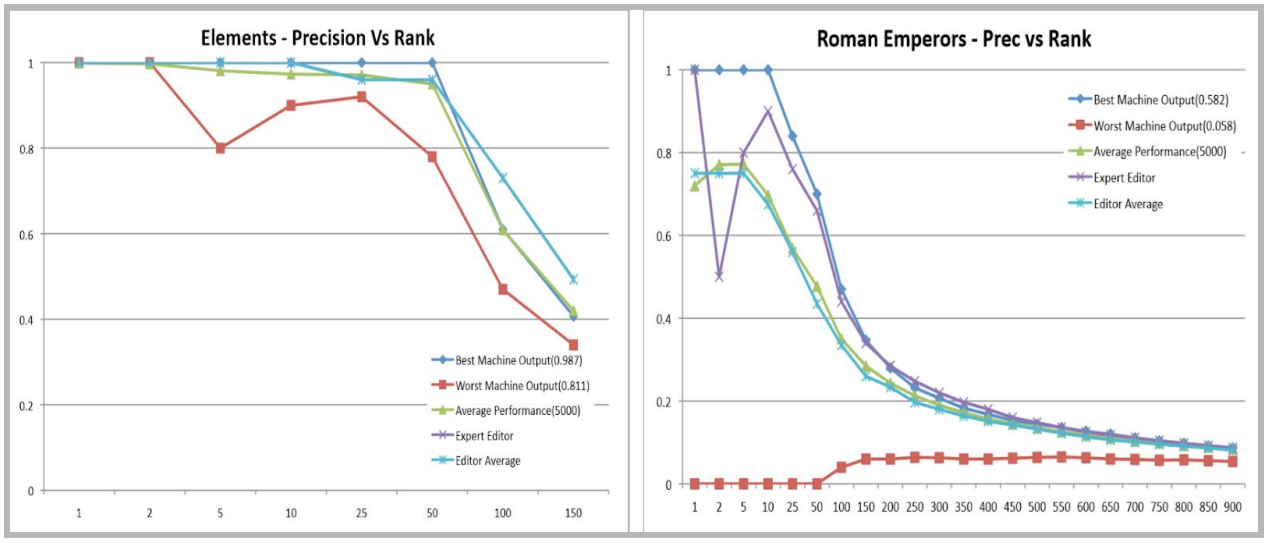
Since this was a first ever approach towards generating task guided taxonomies, there is a vast scope of improvement in the proposed solution. One of the problems that's being highlighted in the paper itself is that HiExpan tends to place synonyms at the same level of taxonomy as they share the semantic meanings and appear in similar contexts. These synonyms tend to make the generated taxonomy less informative, reducing the overall quality.

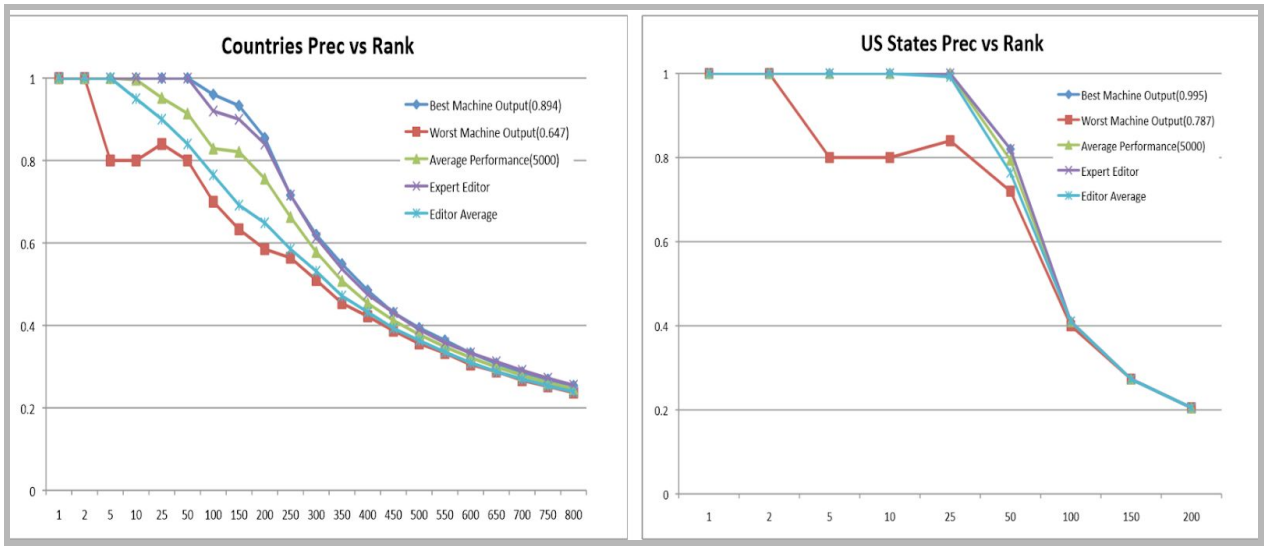
Another thing is, the paper mentions input to be a domain specific corpus. However, as the paper[4] mentions it is difficult to find a domain specific corpus everytime. This difficulty increases as new domains keep coming in frequently. Also, when we wish to generate a taxonomy for a very specific sub domain, availability of that particular specific corpus is rare compared to the parent domain specific corpus. The paper[4] talks about this particular issue and tries to incorporate 'Knowledge + context' in the taxonomy building. If the SetExpan algorithm was modified to take advantage of this hierarchical clustering methodology HiExpan's applicability would increase.

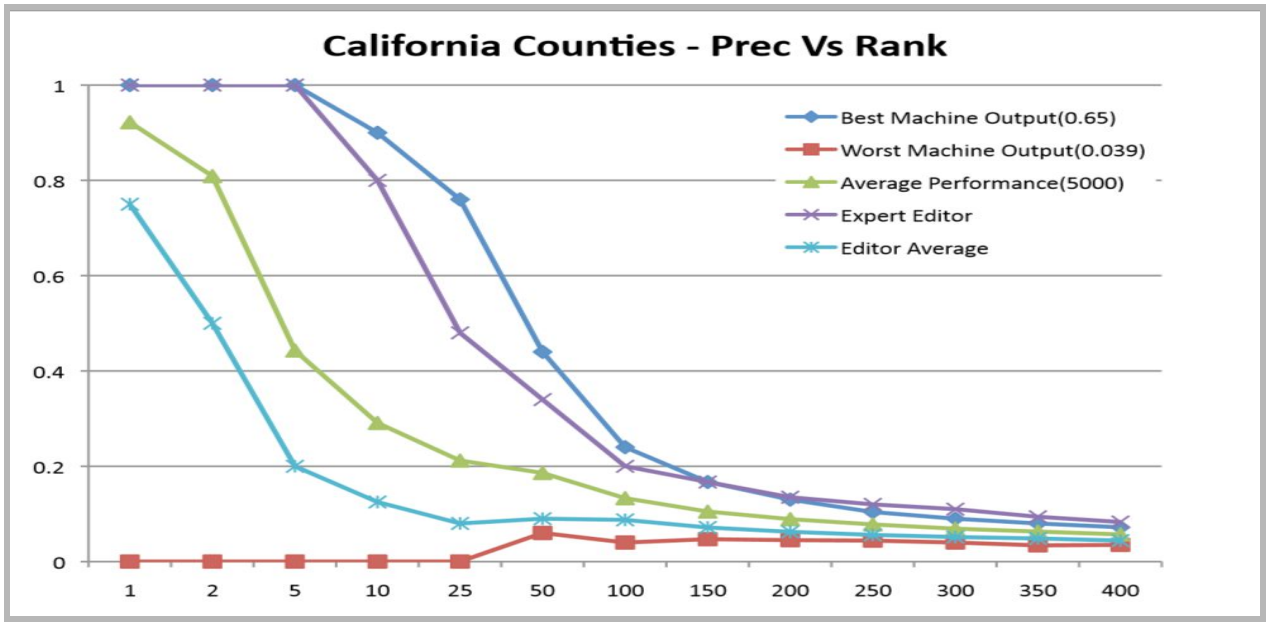
Another scope of improvement lies in selection of proper seeds. The paper[5] studies the effect of choosing the "right seeds" in terms of their composition and how it affects the accuracy of the generated taxonomy. The composition of seeds needs to take care of,

1. Prototypicality - Words which are most representative of a domain. The high frequency of these words includes a lot of variety in contexts which may cause semantic drift in the generated tree.
2. Ambiguity - Polysemy of seed can introduce seed intrusion in the taxonomy. This can be avoided by providing a domain specific corpus like in the case of HiExpan, but when it isn't available this ambiguity needs to be removed.
3. Coverage - Coverage of a seed set for a concept is the amount of semantic space which the seed shares in common with the semantic space defined by the concept.

The impact of these three is shown in the paper[5] using the taxonomy generator in using precision vs rank curves for different datasets. As seen in the graphs, the difference between the selection types varies with different seed compositions.


Elements - Precision Vs Rank
Best Machine Output(0.987)
Worst Machine Output(0.811)
Average Performance(5000)
Expert Editor
Editor Average
Roman Emperors - Prec vs Rank
Best Machine Output(0.582)
Worst Machine Output(0.058)
Average Performance(5000)
Expert Editor
Editor Average


Countries Prec vs Rank
Best Machine Output(0.894)
Worst Machine Output(0.647)
Average Performance(5000)
Expert Editor
Editor Average
US States Prec vs Rank
Best Machine Output(0.995)
Worst Machine Output(0.787)
Average Performance(5000)
Expert Editor
Editor Average



Conclusion:

HiExpan[1] was a novel approach proposed for taxonomy generation. A new concept called task guided taxonomy construction was also introduced in the paper for overcoming the drawbacks of the previous approaches. HiExpan basically combines two approaches for taxonomy generation for constructing a global taxonomy. However, since the paper focused on having a domain specific corpus as input it's applicability is limited. However, for improving this there is an approach that uses modified hierarchical clustering which would increase the applicability of the algorithm. Also, for the set expansion algorithm the quality of initial seeds can be refined using the method proposed in [] to improve the quality of the generated tree.

References:

[1] HiExpan:Task-Guided Taxonomy Construction by Hierarchical TreeExpansion - Jiaming Shen,Zeqiu Wu, DongmingLei, ChaoZhang1, XiangRen, Michelle T. Vanni, Brian M. Sadler, Jiawei Han KDD 18

[2]SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble - Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, Jiawei Han

[3]Weakly-supervised Relation Extraction by Pattern-enhanced Embedding Learning - MengQu, XiangRen, YuZhang, JiaweiHan WWW 18

[4]Automatic Taxonomy Construction from Keywords - Xueqing Liu, Yangqiu Song, Shixia Liu, Haixun Wang KDD 12

[5]Helping Editors Choose Better Seed Sets for Entity Set Expansion - Vishnu Vyas, Patrick Pantel, Eric Crestan CIKM 09

[6]Predictive text embedding through large-scale heterogeneous text networks- J.Tang, M.Qu,andQ.Mei KDD'15.
[7]NetTaxo: Automated Topic Taxonomy Construction from Text-Rich Network. - Jingbo Shang, Xinyang Zhang, Liyuan Liu, Sha Li, and Jiawei Han. WWW 2020.
[8]Automatic Taxonomy Extraction Using Google and Term Dependency - Masoud Makrehchi and Mohamed S. Kamel. WI 2007.
[9]Automatic taxonomy construction from keywords - Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. KDD 2012.
[10]Towards an automatic construction of Contextual Attribute-Value Taxonomies. - Dino Ienco, Yoann Pitarch, Pascal Poncelet, and Maguelonne Teisseire. SAC 2012
[11]Towards automatic generation of query taxonomy: A hierarchical query clustering approach - S.L. Chuang and L.F. Chien. ICDM 2002.
[12]Mining web query hierarchies from clickthrough data - D. Shen, M. Qin, W. Chen, Q. Yang, and Z. Chen. AAAI 2007
[13]Onto Learn Reloaded: A Graph-Based Algorithm for Taxonomy Induction - Paola Velardi, Stefano Faralli, RobertoNavigli CI 2013
[14]A Short Survey on Taxonomy Learning from Text Corpora - Chengyu Wang, Xiaofeng He, Aoying Zhou. EMNLP 2017
[15] Probase:a probabilistic taxonomy for text understanding - Wentao Wu, Hongsong Li, Haixun Wang, KennyQ.Zhu. SIGMOD 2012
[16]Snowball: extracting relations from large plain-text collections - Eugene Agichtein and Luis Gravano.
[17]Distributed Representations of Words and Phrases and their Compositionality - Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S.Corrado, JeffreyDean. NIPS 2013
[18]System and methods for automatically creating lists - S. Tong and J. Dean. US Patent 2008.
[19]Language-independent set expansion of named entities using the web - R. C. Wang and W. W. Cohen.ICDM, 2007.
[20]Long-tail vocabulary dictionary extraction from the web. - Z. Chen, M. Cafarella, and H. Jagadish.WSDM 2016.
[21] Automatic Synonym Discovery With Knowledge Bases. - M. Qu, X. Ren, and J. Han. KDD'17
[22]Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches - Shuming Shi, Huibin Zhang, Xiaojie Yuan, Ji-Rong Wen
[23]https://courses.cs.washington.edu/courses/cse517/13wi/slides/cse517wi13-RelationExtraction.pdf
[24]CERES: distantly supervised relation extraction from the semi-structured web. - Colin Lockard, Xin Luna Dong, Arash Einolghozati, and Prashant Shiralkar. VLDB 2018
[25]https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783558742/1/ch01lvl1sec12/taxonomy-of-machine-learning-algorithms