Enriching Taxonomies With Functional Domain Knowledge

Nikhita Vedula
The Ohio State University
vedula.5@osu.edu

Patrick K. Nicholson
Nokia Bell Labs, Ireland
pat.nicholson@nokia-bell-labs.com

Deepak Ajwani
Nokia Bell Labs, Ireland
deepak.ajwani@nokia-bell-labs.com

Sourav Dutta
Nokia Bell Labs, Ireland
sourav.dutta@nokia-bell-labs.com

Alessandra Sala
Nokia Bell Labs, Ireland
alessandra.sala@nokia-bell-labs.com

Srinivasan Parthasarathy
The Ohio State University
srini@cse.ohio-state.edu

ABSTRACT

The rising need to harvest domain specific knowledge in several applications is largely limited by the ability to dynamically grow structured knowledge representations, due to the increasing emergence of new concepts and their semantic relationships with existing ones. Such enrichment of existing hierarchical knowledge sources with new information to better model the "changing world" presents two-fold challenges: (1) Detection of previously unknown entities or concepts, and (2) Insertion of the new concepts into the knowledge structure, respecting the semantic integrity of the created relationships. To this end we propose a novel framework, *ETF*, to enrich large-scale, generic taxonomies with new concepts from resources such as news and research publications. Our approach learns a high-dimensional embedding for the existing concepts of the taxonomy, as well as for the new concepts. During the insertion of a new concept, this embedding is used to identify semantically similar neighborhoods within the existing taxonomy. The potential parent-child relationships linking the new concepts to the existing ones are then predicted using a set of semantic and graph features. Extensive evaluations of ETF on large, real-world taxonomies of Wikipedia and WordNet showcase more than 5% F1-score improvements compared to state-of-the-art baselines. We further demonstrate that ETF can accurately categorize newly emerging concepts and question-answer pairs across different domains.

ACM Reference Format:
Nikhita Vedula, Patrick K. Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala, and Srinivasan Parthasarathy. 2018. Enriching Taxonomies With Functional Domain Knowledge. In *SIGIR '18: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* , 10 pages. https://doi.org/10.1145/3209978.3210000

1 INTRODUCTION

Human knowledge is inherently organized in the form of semantic, content-specific hierarchies. Large-scale taxonomies such as Wikipedia Categories[1], Freebase [2] and WordNet [6] are crucial sources of structured knowledge useful for various natural language processing applications. However, although these hierarchies are well developed, they are largely generic and laborious to augment and maintain, with new concepts and relations from newly emerging, or rapidly evolving domains such as public health and current affairs. These challenges necessitate the development of automated, scalable techniques to solve the problem of *taxonomy enrichment*, i.e., to augment a taxonomic hierarchy by accurately placing novel or unfamiliar concepts in it at an appropriate location and level of granularity (avoiding links that are too specific, too general, or contextually related but non-ancestral). For example, based on the Wikipedia article on the *Hurricane Harvey* in August 2017, we would like to link it to coherent, specific categories such as *Category 4 Atlantic Hurricanes* and *August 2017 events in the United States* rather than semantically related but non-ancestral classes such as *Hurricane Irma*, or overly general categories like *Hurricane*.

Existing work in the area of automated taxonomy enrichment is still either highly language-specific [47, 48], domain-specific [28] or cannot scale to large taxonomies [3, 21, 43]. Many techniques depend on the unique synset structure specific to WordNet [14, 31, 37], cannot generalize to other taxonomies, and can only identify a single category for new concepts. Even with these limitations, not all attempts at taxonomy enhancement have succeeded [28].

In contrast, we identify that the main challenge for this task is to find a computational measure that can proxy the logic of semantic subsumption, independent of the language and knowledge domain, in order to automatically find good parents for new concepts. To address this fundamental challenge, we propose a combination of a carefully selected set of highly effective graph-theoretic features and semantic similarity based features leveraging external knowledge sources. We show that this combination measure can predict the links to the parents of new concepts with high accuracy. Interestingly, our measure does not require any assumptions on the number of parents for each new concept. Moreover, our evaluation methods are fully automated using publicly available data, and don't rely on domain experts or manual judgment as in [4, 20, 25, 31, 42].

Intuitively, we believe that ETF is similar to how a human would approach the task of adding a new concept to a taxonomy. To complete such a task, one might: (i) find a set of existing concepts that are related to the new one, (ii) rank each concept in the union of ancestors of this set, i.e., the closure via hypernymy/parent relations, and; (iii) link the new concept to a few selected top-ranked parents.

To formalize the intuitive method we just outlined, our proposed approach ETF first constructs a vector representation for new concepts. Here, we use a representation that aggregates two kinds of embeddings formed from the context of the concept. We observed

[1] https://en.wikipedia.org/wiki/Portal:Contents/Categories

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210000

that the embeddings complement each other and so the aggregated vector provides a good measure for semantic similarity, that can identify the nearest neighbors of the new concept in the taxonomy. With this representation, the search for potential parents is restricted to the (small) set of ancestors of these nearest neighbors. Experiments show that on test concepts from the Wikipedia category taxonomy with more than 5 million concepts, on average this set contains a few thousand nodes and covers 83% of the true parents of the test concepts. Finally, we develop algorithms that rank the ancestors in this set, selecting only those above a global scoring threshold to be the parents of the new concept.

Therefore, the key contributions of our work are:

- We develop a novel, fully automated framework, ETF, that generates semantic text-vector embeddings for each new concept. These embeddings allow us to find semantically related concepts in the existing taxonomy, which in turn allows us to extract the ancestors of these related concepts.
- We propose the use of a learning algorithm that combines a carefully selected set of graph-theoretic and semantic similarity based features to rank candidate parent relations. This ranker accurately links new concepts to good candidate parents by ranking the ancestors of their semantic neighbors.
- We test ETF on large, real-world, publicly available knowledge bases such as Wikipedia and Wordnet, and outperform baselines at the task of inserting new concepts. Crucially, our experimental design is easily reproducible, and can be used as a benchmark for future research on this topic.
- Through two case studies, we show that ETF can accurately categorize new concepts from rapidly evolving real-world domains, as well as new questions and answers from Quora[2].

2 RELATED WORK

The problem of automatic taxonomy induction, i.e., effectively reconstructing an entire taxonomy, has received a lot attention in the literature. Supervised and unsupervised machine learning techniques based on co-occurrence analysis [42], clustering [20], graph construction and traversal [25] and distributional similarity [47, 48] have been used to solve this problem. Linguistic pattern-matching based approaches [9, 25, 31, 49] have been employed to discover relations between a term and its hypernyms. Using word embedding based techniques for identifying relations to recreate taxonomies has also gained popularity in recent years [8, 36, 38, 51]. Some of these techniques [8, 36] suffer from low accuracy in taxonomic relation prediction, while others [51] do not generalize to unseen relation instances. The method proposed by Tuan et al. [38] appears to tackle these issues, however the input to its training phase requires hypernym-hyponym pairs to occur in a sentence, which is quite unlikely in the case of Wikipedia-style concept and category names. We also utilize term embeddings as part of our approach to enhance taxonomies (Section 4), however we combine it with well-designed graph-based and semantic features to maximize performance.

There already exist fairly accurate, general-purpose knowledge bases that have been painstakingly curated by experts or via crowdsourcing. Hence, rather than constructing new taxonomies from scratch, our work leverages these existing taxonomies and enhances them with new information. This problem of automatically enhancing the coverage of extant taxonomies has primarily focused on enhancing the WordNet taxonomy. Toral et al. [37] extended WordNet with about 320, 000 named entities and their relations, derived from Wikipedia categories and articles. Widdows [45] developed a method to place an unknown word where its neighbors are most concentrated, an idea also leveraged by our work. But his work used part-of-speech tagging to find the nearest neighbors, which is unlikely to work with the Wikipedia-style concept names that ETF can handle. WordNet augmentations have also been proposed for the domains of technical reports [41], medicine [4, 7], and architecture [1]. However, many of these efforts suffer from low accuracy, require part-of-speech tagging, and depend on the category structure peculiar to WordNet. Thus, they cannot easily generalize to other taxonomies, unlike our approach. These works also need human judges for evaluation, while ETF does not. Task 14 of SemEval 2016 [15] is based on extending WordNet with concepts from various domains such as religion, law and finance. We evaluate the performance of ETF on this task in Section 5.

Efforts similar in motivation to our work have attempted to extend a generic taxonomy with domain specific knowledge. Ancestor-descendant relationships and hypernym patterns have been extracted from large text corpora such as Wikipedia [9, 33], to enhance taxonomies. Yamada et al. [47, 48] augmented the Japanese Wikipedia and WordNet with new Japanese terms. They first found similar words from the Wikipedia database, scored the hypernyms of these words, and selected the top-scored hypernym as the output. However, their scoring technique is heavily dependent on verb-noun dependencies found in the Japanese language, that are not usually seen in English. These works also attach the new word to a single parent, and require human judges for evaluation.

Recently, the related problem of *knowledge graph completion*, i.e., predicting relations between entities using supervision from an existing knowledge graph, has received attention in the literature [3, 21, 43]. These techniques predict missing links in knowledge graphs by learning entity and relationship embeddings, based on the idea that the relation between two entities corresponds to a translation between their embeddings. We compare ETF against a state-of-the-art link prediction method, *TransR* [21], in Section 5. However, we note that the processing time of these methods is of the order of days for large taxonomies (e.g., the Wikipedia category taxonomy).

3 PROBLEM FORMULATION

We now formally define our taxonomy enrichment problem. A *taxonomy* $T = (V, E)$ is a directed, acyclic graph (DAG) where the set of vertices or nodes V consists of all its hierarchically organized entities and categories, and the set of edges E represents the node relationships. The direction of an edge in the graph hierarchy is from a specialized node to a more generic node that subsumes it. *Entities* in T are designated by nodes that do not have any incoming edges to them, i.e., they have an in-degree of 0. Since these are at the lowest or most specialized 'level' in their respective sub-hierarchies in T, we also refer to them as *leaf* nodes. We use the term *concept* to refer to any kind of node, leaf or otherwise, in T. The *ancestors* of a node v in T are the set of nodes $A(v)$ in T reachable from v. v is thus considered a *descendant* of all nodes belonging to $A(v)$.

[2] http://quora.com

The *k-hop neighborhood* of node v is the set of all nodes that are reachable from v via a path of length at most k in T. The immediate ancestors of v, reachable in a single hop, are called its *parents* or *hypernyms*, whereas v is called a *child* or *hyponym* of its parents.

Though the taxonomy graph *should* be a DAG, it may not be initially free of cycles, due to human error, or from unifying collaborative efforts from disparate sources while constructing the taxonomy. Since they can lead to hierarchical inconsistencies, we remove cycles from our taxonomy via known methods [34, 35].

We assume that existing concepts have some text associated with them. For example, in the Wikipedia category taxonomy, the Wikipedia pages provide the text. However, if the taxonomy was created from a document corpus, an aggregation of the context around each mention of the concept can be used as the associated text for that concept. The inputs to our problem therefore are:

(1) A corpus of unseen, unlabeled, unstructured text documents containing a set of new concepts X. Ideally, each document would provide a definition for exactly one new concept. This definition can be as short as a couple of sentences. However, it is more likely that each document may refer to one or more of the new concepts, without specifically defining these new concepts. Our approach can handle either scenario, though we focus primarily on the more difficult latter case. Here we expect that the corpus contains many (i.e., at least 10) references to each new concept.

(2) A taxonomic hierarchy of categories and entities, T.

(3) A corpus of text documents D associated with the concepts present in T. Note that a document can be as small as a single sentence of text. These documents need not be formal definitions of the concepts in T, and may even be automatically created.

Each concept $x \in X$ must be checked to determine if it is already present in T. If not, then a solution inserts x into T by outputting a set of semantically appropriate parents in T for x.

Discussion: We emphasize that our primary focus in this work is finding the candidate parents for x, rather than the first step of verifying whether x is already present in X. We also note that our problem formulation implicitly assumes that each new concept will be inserted into the taxonomy T as a *leaf* node. This follows since we find a set of parents for a new concept, but do not attempt to find its children that may exist in the taxonomy. The decision to restrict our search to parents was taken due to practical reasons:

(1) Taxonomies tend to have many more leaves than 'non-leaf' concepts. Hence, finding candidate children is often prohibitively expensive. A new concept may have a few thousand candidate parents, but potentially millions of candidate children.

(2) Large knowledge bases (e.g., WordNet and Wikipedia) have well developed categories. Thus, *most* new concepts being added are likely leaves or will eventually be extended top-down to become roots of sub-graphs, so leaf insertions are, by frequency, *the most important* case to consider.

(3) We acknowledge that a bottom-up procedure, in which the taxonomy is extended by inserting new categories above a set of leaves is also possible. Using ETF, it *is* possible to insert new categories into the taxonomy, by linking them to appropriate parents as well as children. However, beyond the cost of finding and examining all candidate children, there are other drawbacks to this approach: (i) category insertions add a temporal dimension, as the order in which entities and categories are inserted matters. This temporal aspect not only complicates the experimental set-up, but also the evaluation and presentation of the results; (ii) category insertions can introduce cycles into the taxonomy, which must then be detected and resolved.

We thus focus only on leaf (i.e., entity) insertions in this work.

4 THE ETF FRAMEWORK

In this section, we describe our framework, ETF (Figure 1). We first learn a representation for the new and existing concepts in the taxonomy (Section 4.2). For each new concept, we leverage its representation to identify the entities in the taxonomy to which it is most similar. We then narrow down the search for the new concept's potential parents to the set of ancestors of the similar entities. Using a combination of graph-theoretic and semantic features as a proxy for semantic subsumption (Section 4.3), our framework filters and ranks these ancestors, and adds appropriate links to the taxonomy.

4.1 Finding Concepts and Taxonomic Relations

This is the first step of our algorithmic pipeline. We acquire the entities and categories from the given taxonomic structure and utilize their ancestor-descendant relationships to construct a DAG T as mentioned in Section 3. Generally, all entities (leaf nodes) and many categories (non-leaf nodes) in T are associated with text documents, which make up the corpus D. The next task is to obtain the novel concepts to be integrated into T. For instance, these could be the name of an emergent disease, a new organism species, or a recently passed law. In many cases, what is available to us is only a text corpus such as news reports, laboratory records or research articles containing descriptions of unfamiliar concepts or ideas. Off-the-shelf Named Entity Recognition algorithms [11, 24] can be applied to locate and extract named entities from the text. This step must be followed by segregating the novel entities from

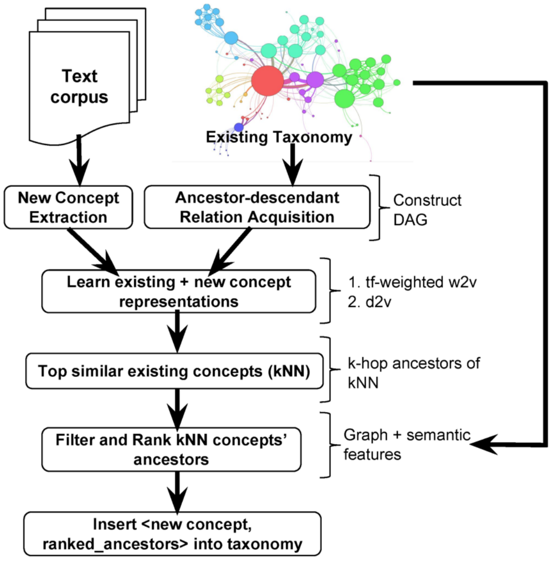
Text corpus
Existing Taxonomy
New Concept Extraction
Ancestor-descendant Relation Acquisition
Construct DAG
Learn existing + new concept representations
1. tf-weighted w2v
2. d2v
Top similar existing concepts (kNN)
k-hop ancestors of kNN
Filter and Rank kNN concepts' ancestors
Graph + semantic features
Insert <new concept, ranked_ancestors> into taxonomy

Figure 1: An overview of the pipeline of our ETF framework.

those already present in T, which, as mentioned, is a challenging problem in its own right and not the focus of this work. However, we do perform a preliminary investigation of the capability of our framework to verify that these new concepts indeed do not already occur in the current taxonomy (Section 5.1.1). For later steps though, we assume that the new concepts to be incorporated into the extant hierarchy are precisely known beforehand.

4.2 Learning Concept Representations

After acquiring the relevant concepts and their relationships, we propose to learn a meaningful, text-based embedding for the existing and new concepts. Recent literature (e.g. [8, 16, 39, 40, 51]) has seen an extensive use of learning and employing high-dimensional word embeddings for diverse applications. We build upon the highly effective Skip-gram variant of word2vec embeddings [23]. It minimizes the log loss of predicting the context of an instance using its embedding as input features. Formally, let (i, c) be the set of pairs of entity i and context c. The loss function is then:

$$-\sum_{(i,c)} \log p(c|i) = -\sum_{(i,c)} \left(\mathbf{w}_c^T \mathbf{e}_i - \log \sum_{c' \in C} \exp(\mathbf{w}_{c'}^T \mathbf{e}_i)\right)$$

where $\mathbf{w}$'s are the parameters of the Skip-gram model, $\mathbf{e}_i$ is the embedding of entity i and C is the set of all possible contexts. Under Skip-gram, for each training pair (i, c), the term i is the current term (entity) whose embedding is being estimated, and the context c is made up of the words surrounding i within a fixed window size.

We first replace all occurrences of entity names (i.e., the specific, potentially non-unique, noun phrases that are associated with that entity) in the corpus D of existing text documents. This can be done by disambiguating the noun phrases [22] to the correct entity, and representing the entity by a canonical unique term. Note that, in the case of Wikipedia, these links have already been established by human annotators, so the disambiguation step is not necessary.

We then train a doc2vec (distributed memory or DM version of Paragraph Vector) [19] model with negative sampling [23] on D, in the joint space of terms and entities (each represented by a canonical term), that gives us vector representations for the documents in D. Note that this is different from the past works on embeddings that exclusively work in the space of terms, as the step of replacing all the (possibly non-unique) set of noun phrases that refer to a specific entity by a canonical identifier removes a great deal of ambiguity from the corpus. Doc2vec has a similar loss function as specified above except for an additional 'paragraph token' that is pre-pended to the beginning of the context. The DM mode of doc2vec simultaneously learns word2vec word vectors and document vectors in the same space during training, thereby enabling easy aggregation of these vectors.

To get the representation of an entity, we add a *tf*-weighted sum of the word2vec embeddings of its context terms to the doc2vec representation of its associated document. The intuition behind this representation is that word embeddings effectively capture the semantic relationships between individual words in a document, while document embeddings summarize the overall semantics of their constituent words. By aggregating the two vectors, we hope to maintain the representational effectiveness and non-sparsity of doc2vec, while also preserving individual word semantics. Furthermore, we find that the document representation constructed from word2vec is biased towards highly frequent terms or those that express functions. However, the doc2vec representation is more affected by low frequency, content-rich words, since the more frequent words are likely to be chosen as negative samples. We empirically demonstrate the benefits of aggregating these two kinds of embeddings in Section 5 (Figure 2).

After creating embeddings for the existing concepts in T, we next learn representations for the new concepts to be inserted into T. This is essentially the task of inductive learning [44], which generalizes an existing learning model to produce representations for unseen data items. However, it is non-trivial to achieve this in an unsupervised setting. Prior work has done this by either incrementally re-training the learning model [27] which is expensive, or using semi-supervised techniques based on neighborhood attributes [44, 50] to learn embeddings of unobserved instances. To accomplish this in an unsupervised manner, we first generate a context c for each new concept, composed of the frequent terms based on their *tf*-score, aggregated from all documents in the corpus where the new concept occurs. We infer a doc2vec embedding using c via gradient descent, by keeping the current doc2vec model constant, as proposed in [19]. We then approximate the embedding of the new concept by adding the *tf*-weighted sum of the word2vec embeddings of its context terms to its inferred doc2vec embedding.

4.3 Filtering and Ranking Potential Parents

With the embeddings of the new and existing concepts in hand, we now must determine the best parents for each new concept within T. To this end, for each new concept, we find the k-nearest neighbours (kNNs) that are most similar to it in the vector embedding space. We hypothesize that the best parents for the new concepts are highly likely to be in common with the ancestors of these kNNs. What remains now is to rank the candidate parents for the new concept.

For this pupose, we build a learning-to-rank model that uses topological features from the taxonomy's graph structure, and semantic features derived from the text. Specifically, we used LambdaMART [5] with a set of training examples consisting of relevant and non-relevant parents of the existing concepts in T. Note that we tried seven other learning-to-rank models (e.g., SVMRank [13]), but found that LambdaMART gave the best performance. The test set consists of potential parents of a set of new concepts, disjoint from the training examples, to be ranked. As mentioned earlier, these candidate parents are the ancestors of the kNNs of the new concepts. 'Relevant' or positive training examples are therefore all those categories that are parents of existing concepts belonging to the training set. 'Non-relevant' or negative training examples are categories that have been randomly sampled from the ancestors of the kNNs of the new concept. We take an equal number of positive/negative training examples for each concept.

We studied fifteen topological or graph-based and semantic features, before selecting a subset of six features that provides a good coverage of various structural and content-based properties, coupled with better performance of the learning-to-rank model. The actual values of the features can either be a score defined for a pairwise property (usually the pair of a new concept and its potential parent), or for an individual property (of the potential parent). The topological features we studied include measures relating to graph centrality, and path traversals over the taxonomy DAG. We also studied semantic features based on the contextual difference between new concepts and their prospective parents, their co-occurrence, contextual term overlap, term overlap with respect to certain parts-of-speech, and lexico-syntactic patterns such as hypernym patterns or sibling patterns [9, 31]. Below, we describe the features most beneficial for our ranking model.

4.3.1 Graph-based Features

We found three graph-based features to be particularly effective in ranking candidate parents: Katz similarity, random walk betweenness centrality and an information propagation measure. The first two are based on an undirected graph obtained by taking the nearest neighbor nodes and their ancestors up to a certain depth. We then take all the directed edges between these nodes from the taxonomy and insert them into the undirected graph.

Katz Similarity: Intuitively, there should be many short paths between the new concept and its correct parents. Katz similarity [17] captures this intuition and hence turns out to be a highly discriminative indicator for semantic subsumption. We compute the Katz similarity between each nearest neighbor x and potential parent p, and average them for each potential parent, as,

$KS(x,p) = \sum_{l=1}^{l_{max}} \eta^l \cdot |paths_l(x,p)|$, where $|paths_l(x,p)|$ is the number of paths of length $l \leq l_{max}$ between x and p, and η is an attenuation parameter ($0 < \eta < 1$) that ensures that shorter paths are weighted higher.

Random Walk Betweenness Centrality: A good parent should generalize the most similar neighbors of the new concept well. One way to capture this is by considering random walks from the neighbors and measuring the betweenness centrality of the candidate parents based on these random walks. A general random walk betweenness centrality measure [26] roughly measures how often a node is traversed by a random walker going from any node in the network to another. Hulpus et al. [12] proposed a focused random walk betweenness centrality measure that focuses on the paths between all pairs of a pre-defined subset of nodes. Our feature is akin to this focused measure, where the pre-defined subset is the set of neighbors of the new concept.

Information Propagation Score: One problem with considering undirected graphs for concept generalization is that it can result in topic drift and hence, the resultant measure becomes noisy. To address this we consider another feature, namely an *information propagation score*, that propagates the weight from the neighbor nodes upward along directed edges to more general concepts, following the directional traversal on the hypernym edges.

Consider the set of nearest neighbor entities for the new concept, where each entity has a weight associated with it, i.e. the similarity between itself and the new concept to be inserted into T. These weights are then propagated upwards towards the ancestor nodes in T. This propagation ensures that highly central parent nodes lying on many neighbor-to-root paths accrue large weights. To avoid over-generalization, each intermediate node decays the weights by a multiplicative factor $(1 - \delta)$, which we call the *decay factor*.

However on noisy, real-world taxonomies such as Wikipedia categories, additional issues may arise. First, there can be a great disparity in the number of parent categories $P(v)$ of each node v. This causes the parents in $P(v)$ to obtain low weight values if $P(v)$ is large, or large values if $P(v)$ is small, if the propagated weights are uniformly split among all parents. We thus introduce a parameter $\alpha < 1$ that enables us to propagate a proportion $\frac{1}{p(v)^\alpha}$ of the weight at each step. Second, while most edges in T are likely to be between nodes at similar levels of granularity, some may be direct connections between highly specialized and highly generic concept nodes. This can cause a generic node to get a higher weight than its counterparts at similar levels. We thus additionally penalize highly generalized nodes far away from leaf nodes by a factor β.

Let $p(v)$ be the number of parents of node v in $T(V,E)$, and N_x be the set of nearest neighbor entities of a new concept x. The initial weight of each entity v in N_x is given by $w_0(v)$, which is the cosine similarity between v and x in the embedding space, and 0 otherwise. Let $\mathrm{ld}(v)$ be the length of the longest path from a leaf node in T to v. The information propagation score $IP(v)$ is defined as the total weight passing through v from the leaf level, given by:

$$IP(v) = \begin{cases} w_0(v), & v \in N_x. \\ \sum\limits_{(u,v)\in E} \frac{(1-\delta)IP(u)}{p(u)^\alpha e^{\mathrm{ld}(v)\beta}}, & v \notin N_x; u, v \in V. \end{cases}$$

4.3.2 Semantic Features

In addition to the graph features, we found the following semantic features to be highly discriminative:

Ancestor-Neighbor Similarity: For each novel concept, we compute the pairwise term overlap between the text document linked to its potential parent under consideration, and that associated with each of its nearest neighbor entities. We then take the average of these overlap values. The importance of this feature stems from the fact that a good ancestor should generalize the properties of as many entities highly similar to the target concept as possible. For this feature, we only consider those ancestors or parents in the taxonomy DAG T that have text associated with them.

New concept-Ancestor Similarity: Since a satisfactory parent of a new concept is a good generalization of it, it is quite likely to have similar text as the new concept itself, i.e. high textual overlap with the new concept. This feature thus computes the Jaccard similarity between the occurrences of textual terms in the document associated with the new concept, and those associated with the category or ancestor document. As earlier, we only consider those ancestors in T that have text associated with them.

Pointwise Mutual Information (PMI): A parent of a concept has a high probability of co-occurring or being mentioned together with it. Yang and Callan [49] have shown this feature to be highly successful in indicating semantic relations between terms. PMI measures the co-occurrence of a new concept x and its potential parent p via the pointwise mutual information between them:

$PMI(x,p) = \log\left(\frac{num(x,p)}{num(x) \cdot num(p)}\right)$, where $num(t)$ (or $num(t_1, t_2)$) is defined as the number of occurrences of a particular term t (or co-occurrences of a pair of terms), either in a set of sentences, or documents which can be from a large corpus such as Wikipedia or the web. We tested our approach by computing the PMI using two such corpora of documents. First, we investigated the PMI between each new concept and each of its prospective parents by checking their co-occurrence in Wikipedia. However, many peculiar concept and category names do not occur in text documents, hence we could not gain a performance boost with this feature. The second kind of PMI that we compute is by checking the co-occurrence of the new concepts and their potential parents in all pages on the web. We thus compute the value of $num(t)$ by querying the Bing Search API[3] to find the number of search results or pages on the web containing the respective term t. Evidently, we only consider those potential ancestors in T that have at least one web page in which they occur. Henceforth, all references to the feature 'PMI' refer to the PMI value computed using the Bing Search API.

Once the ancestors have been ranked in order of their relevance to the new concept, we connect the new concepts to their 'r' top-ranked ancestors and evaluate the resulting taxonomy (Section 5).

4.4 Parameter Tuning

In this section, we describe how various parameters were tuned.
Embedding Parameters: We trained 200 dimensional embeddings using the gensim implementation [29] of doc2vec. We used its distributed memory version since it was found to outperform other variants [19], and a context window size of 10.
Number of Neighbors and Depth of Ancestors: These values are selected according to an accuracy/computational cost trade-off. We choose the top 50 nearest neighbor entities to the new concept, whose ancestors we want to evaluate, since we found that additional noise is added to the set of potential ancestors as this value increases. Further, rather than considering *all* the ancestors, we only take into account those ancestors that are reachable from the nearest neighbor entities in at most 3 hops in T. We found that ancestors up to three levels above the neighbors of the new concept span a large portion of T without being too generic or too specific.
Parameters of Graph Features: For the parameters α, β and δ of the information propagation feature, we perform an exhaustive grid search on a sample of our training dataset, and find the best empirical performance with $\alpha = 0.75$, $\beta = 0.005$ and $\delta = 0.005$. We assign the value of the probability parameter q used for computing the random walk betweenness centrality equal to the value of α, i.e. 0.75. For the Katz similarity feature, we use $\eta = 0.2$.
Predicting the Number of Parents: We learn a global ranking threshold based on the rank scores received by the correct, ground truth parents of the training set. The new concept is thus connected to all those parents receiving a rank higher than the threshold value from the ranking model. We found that we could predict the *exact* number of parents of about 70% of the new concepts using a normalized rank threshold of 0.4, and the correct number of parents ±2 for about 87% of the new concepts. We also tried a local, concept-specific ranking threshold, based on the level (i.e distance from root) of the top ranked parents output by our ranker. However, this idea did not yield as good results, possibly due to the manual curation of Wikipedia categories, or the presence of many edges from very general to highly specific nodes in the taxonomy. Hence we report all our results using this global normalized ranking threshold.

5 EVALUATION

We tested ETF on two large-scale general-purpose taxonomies, Wikipedia Categories and WordNet, which we detail next.

5.1 Enhancing Wikipedia Category Taxonomy

To set up the experiment, we extract the entities and categories of the Wikipedia category hierarchy available via DBPedia [10], formatted as per the Simple Knowledge Organization System (SKOS). Wikipedia articles that are not category pages (containing lists of categories and sub-categories) are considered as entities. Relations between categories and entities are represented by the 'skos:broader' relationship. We then construct a DAG T, with the entities as leaf nodes and edges between entities and categories or among categories. This taxonomy exceeds 5 million concepts.

We now outline the procedure we adopt to generate the training and testing data for our approach. First, we use the hyperlink structure of Wikipedia to identify all noun phrases that are ambiguous, i.e., they have more than one concept associated with them as per the Wikipedia ontology, and which occur at least 10 times (i.e. $support \geq 10$). For example, Apple the company and apple the fruit are different entities that share the noun phrase "apple". We randomly sample 6000 of these ambiguous noun phrases. For each phrase, we randomly sample one sense (e.g., Apple, the company) and include it in our test set of new concepts to be inserted into the Wikipedia category taxonomy.

To ensure that the new concepts are independent of the existing ontology and do not have any presence within it, we also filter all Wikipedia pages that are linked to the sampled ambiguous entities. This process resulted in the removal of about 11% of the total Wikipedia pages. In the remaining 89% of the pages, we replace the occurrences of each concept or entity by its unique canonical identifier. This modified set of articles is used as textual input to train our embedding model (Section 4.2). The mentions of each new concept from the 11% of pages that were deleted are used to form contexts for the new concepts. To facilitate reproducibility, we provide (see https://github.com/vnik18/Taxonomy) the: (i) Wikipedia version we used; (ii) list of concepts, and; (iii) preprocessing scripts.

5.1.1 *Evaluating Concept Representations*

We begin by testing the quality of the aggregated vector embeddings of the new concepts, by checking how well separated they are from the existing entities' embeddings, i.e., all Wikipedia entities excluding the pages on the new concepts. The y-axis of Figure 2 shows the percentage of new concepts whose pairwise cosine similarity with the taxonomy entities (averaged over all entities) is greater than or equal to the cosine similarity threshold on the x-axis. We evaluate the performance of the three kinds of embeddings from Section 4.2 in distinguishing the new concepts: (i) tf-weighted word2vec embeddings, (ii) doc2vec embeddings, and (iii) summing these two kinds of embeddings. Aggregating the doc2vec and tf-weighted word2vec embeddings gives us a 2-5% improvement in the quality of separability of the new concepts. From Figure 2, we observe that only about 28% of the novel test concepts have an average pairwise similarity of ≥ 0.4 with existing concepts using the aggregated embedding. In other words, more than 70% of the novel test concepts are well-separated from the prevailing entities in the embedding space at a cosine similarity threshold of 0.4.

5.1.2 *Evaluating Our Ranking Model*

Baselines: These include feature-based variants of our ETF ranker, and a state-of-the-art knowledge graph completion technique, *TransR*:

[3] https://azure.microsoft.com/en-us/services/cognitive-services/

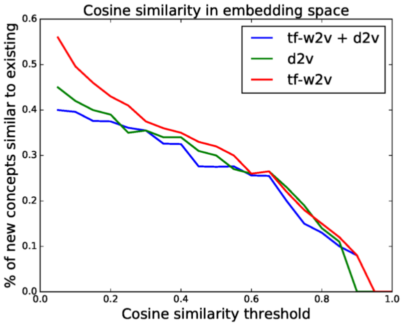

Figure 2: Investigating the quality of ETF embeddings in identifying new concepts from prevailing taxonomy entities

(1) Random: It connects each new concept to a fixed number of randomly selected parents in T, equal to the average of the actual number of parents for the new concepts in ground truth (*ranker-rand*), i.e. 6 parents in case of Wikipedia.
(2) Text similarity: Each new concept is linked to the top 6 potential parents with the highest text-based Jaccard similarity between the parents and the new concept (*ranker-textsim*).
(3) Graph features: This baseline (*ranker-gr*) only makes use of the features based on the taxonomy DAG properties as defined in Section 4.3.1, as input to the learning model.
(4) Semantic features: We train two kinds of rankers using only semantic features (Section 4.3.2). The first, *ranker-sem-noPMI*, is trained on the textual features excluding PMI, and the second, *ranker-PMI*, is trained only on the PMI feature.
(5) TransR: We use the TransR [21] based concept embeddings and the embedding of a single relationship type *parent*, to predict the top 6 parents of each new concept (*ranker-TransR*).

Performance Measures:

(1) Precision@r: We report the precision of each approach, micro-averaged over each new concept from the testing set. This is the number of correctly identified parents in the top 'r' ranked parents, divided by the number of parents retrieved.
(2) Recall@r: The recall is the (micro-averaged) number of correctly identified parents in the top 'r' ranked parents, divided by the correct number of parents of the new concept.
(3) NDCG@r: We evaluate the performance of each approach by measuring the Normalized Discounted Cumulative Gain (NDCG). For a new concept x that has P parents $p_1, p_2, ..., p_P$, the discounted cumulative gain (DCG) at rank r and the ideal discounted cumulative gain (IDCG) at r are defined as: $DCG@r = \sum_{i=1}^{P} \frac{1}{\log_2(rank(p_i)+1)}$; $IDCG@r = \sum_{i=1}^{P} \frac{1}{\log_2(i+1)}$ where rank(p_i) is the rank of parent p_i in the ranked list of parents deduced by the approach. $NDCG@r$ is the $DCG@r$ divided by the $IDCG@r$. Since NDCG is defined for a single concept or document, we report the micro-averaged NDCG which is the average of $NDCG@r$ over all test concepts.

Ranker Comparison Results: Table 1 displays the performance of our approach against baselines on inserting 6000 new concepts into the Wikipedia category taxonomy. We report the values of precision, recall and NDCG by considering the top r parents, where r varies for each new concept based on the number of parents whose rank scores are above the rank threshold. Trained on a combination of graph and semantic features excluding PMI, *ranker-ETF* is able to achieve an NDCG of 0.73 and F1-score of 0.7. This is a 6% improvement in F1-score and a 12% improvement in NDCG over using TransR-based embeddings in *ranker-TransR*. Including PMI, *ranker-ETF-PMI* achieves about 6% improvement in recall and 1.5% in NDCG, but a drop in precision. We reiterate here that we only use those ancestors of the nearest neighbor entities of the new concepts that are reachable to them in at most 3 hops, as prospective parents of the new concepts. This restriction allows us to cover about 83% of the true parents of the new concepts, eliminating the remaining 17% of the parents belonging to ground truth from the reach of our technique. That is, our approach has a performance upper bound equal to the proportion of coverage of true parents, i.e. 83%. Judging by the performance of the ranker learning only from graph-based features (*ranker-gr*) and that learning only from semantic features (*ranker-sem-noPMI*), we conclude that both kinds of features lead to the efficacy of our approach.

Table 1: Performance results on the Wikipedia taxonomy.

Approach	NDCG@r	Precision@r	Recall@r	F1
ranker-rand (r=6)	0.104	0.143	0.211	0.17
ranker-textsim (r=6)	0.328	0.293	0.381	0.331
ranker-gr	0.367	0.451	0.46	0.455
ranker-sem-noPMI	0.46	0.48	0.55	0.513
ranker-PMI	0.54	0.46	0.524	0.49
ranker-TransR (r=6)	0.612	0.69	0.61	0.647
ranker-ETF	0.73	**0.72**	0.67	0.7
ranker-ETF-PMI	**0.745**	0.68	**0.73**	**0.704**

Table 2 shows the average number of parents our method was able to correctly detect ('Hits'), and the parents that were not in the ground truth but were reachable within k-hops from the true parents. For instance, they could be parents of the true parents, or parents of the parents of the true parents. We are able to correctly predict about 67% of the parents of the new concepts, and an additional 8% of reasonably good parents since they are reachable from the true parents in ≤ 4 hops. But they are slightly more generalized than needed. We completely miss 25.34% of the parents, of which 17% are missed due to the 83% coverage limitation of our algorithm.

Computational Cost and Feature Importance: Our framework inserts one new concept per minute on average. The bottlenecks are the random walk betweenness centrality, and the PMI API calls, as the remaining features can be computed in seconds. However, in principle, a fast batch PMI implementation [18] can replace individual API calls, so the main bottleneck is the random walk betweenness centrality feature.

The three most important features, based on individual performance, are information propagation, random walk betweenness centrality, and Katz similarity. For Katz similarity, we also experimented with a directed variant, and found that it was more correlated with information propagation than the undirected variant, with Pearson correlation values of 0.4 and −0.037, respectively. However, there was negligible change in performance using directed vs. undirected Katz in conjunction with the other features.

Table 2: Average percentage of top 'r' parents reachable from predicted parents of new concepts within k hops.

Hits	1-hop	2-hop	3-hop	4-hop	5-hop	Misses
0.664	0.684	0.729	0.738	0.746	0.746	0.2534

Table 3: Performance on the WordNet taxonomy for the SemEval 2016 Task 14, using the measures of Lemma Match, Wu&Palmer similarity (Wu&P), Recall and F1 score.

Approach	Lemma Match	Wu&P	Recall	F1
Random synset	0	0.227	**1**	0.37
FWFS	0.415	0.514	**1**	0.679
MSejrKU System 2	0.428	0.523	0.973	0.68
ranker-ETF	0.42	0.473	**1**	0.642
ranker-ETF-FWFS	**0.5**	**0.562**	**1**	**0.72**

5.2 Evaluation on SemEval Task Benchmark

We further evaluate ETF on the WordNet 3.0 onotology [6], on the dataset of new concepts provided as part of the SemEval 2016 Task 14 [15]. It was specifically constructed to span a wide range of domains, including online resources such as Wiktionary and other glossary websites. This dataset contains 1000 concepts, split into a training set of 400 and a testing set of 600 concepts, which are either nouns or verbs. The task consists of choosing one of two operations for each new concept; (i) attach, where the novel concept needs to be added as a new synset in WordNet, and (ii) merge, where the novel concept needs to be added into an existing synset. For each concept, a textual definition of a few sentences and a single correct parent for it has been provided as part of the task. ETF chooses as the parent for each new concept the top ranked ancestor by our ranker. The design of our algorithm does not permit us to 'merge', we only perform the 'attach' operation for every unseen concept.

5.2.1 *Experimental Setup*

At the outset, we establish a DAG of the WordNet taxonomy, consisting of all the existing entities or leaves, connected to their respective categories via hypernym-hyponym relationships. Each WordNet synset corresponds to a single DAG node. We use the definition of the test concepts provided by Task 14 as their context. We infer the embedding for each new concept as in Section 4.2, from the doc2vec model trained on Wikipedia articles, after making sure that any article pages on the new concepts, and pages that link to them have been removed. We then find the existing Wikipedia entities that are most similar to the new concept in the embedding space. However, one issue that arises here is that these nearest neighbor entities and their hypernyms are part of the Wikipedia category taxonomy, whereas we want to link the new concept to its most likely WordNet hypernyms. Hence, we employ the YAGO ontology [32] to link the Wikipedia neighbor instances to the corresponding instances at the leaf level of the WordNet hierarchy. However, all Wikipedia neighbor entities may not have corresponding WordNet counterparts. We eliminate all those nearest neighbor entities from consideration which do not map to a WordNet instance. We finally compute the graph-based and semantic features as in Section 4.3, from the constructed WordNet DAG and embedding model, to rank the potential hypernyms of the new concepts.

5.2.2 *Baselines and Performance Measures*

To evaluate our approach, we use as baselines those used in Task 14, and the winning system of the task, *MSejrKU System 2* [30]:

(1) Random synset: It attaches the new concept to a random WordNet synset, with the same part-of-speech as itself.
(2) First-Word-First-Sense (FWFS): It links the new concept to the first word in its definition with the same part-of-speech as itself, stemming from the grouping of glosses in WordNet.
(3) MSejrKU System 2: This was the system that won Task 14.

We assess our method using the measures defined in Task 14:

(1) Accuracy (Wu&Palmer Similarity): Wu and Palmer [46] defined the semantic similarity between the predicted parent x_1 and the true parent x_2 as: $Wu\&P(x_1, x_2) = \frac{2*depth_{LCA}}{depth_{x_1}+depth_{x_2}}$ where LCA is the Least Common Ancestor of x_1 and x_2, and $depth_{x_i}$ refers to the depth of x_i in the WordNet hierarchy.
(2) Lemma Match: It measures the proportion of test concepts for which an algorithm selects a synset with at least one word in common with the correct synset for that concept.
(3) Recall: This is defined as the percentage of test concepts for which an output ancestor was identified by an algorithm.
(4) F1 score: Harmonic mean of the Wu&Palmer score and recall.

5.2.3 *Results*

As mentioned earlier, though the task permits two operations of 'attach' and 'merge', we only perform the 'attach' operation. Just 'attaching' every new concept to its own independent synset causes the ceiling (upper bound) on the achievable F1-score to be about 0.989, with precision 0.98 and recall 1. Table 3 exhibits the performance of ETF on the SemEval task on WordNet. We evaluate two versions of our approach, (i) a ranker trained on the same set of features as described in Section 4.3 (*ranker-ETF*), and (ii) the same ranker using an additional binary feature of the FWFS property (*ranker-ETF-FWFS*): if a prospective parent of an unseen concept is the first word of its definition with the same part of speech as itself, the value of this feature is 1, otherwise it is 0. We observe that our ranker without the FWFS feature gives an F1 score of 0.642, however with the FWFS-based binary feature, ETF's F1 score outperforms the best system of this task by 0.04. This is particularly impressive given that very little improvement compared to FWFS (0.001 in F1 score) has been recorded on this benchmark in the past.

We note that FWFS is a strong feature for ranking parents in the WordNet taxonomy, since the organization of word glosses in WordNet ensures the presence of the word expressing the ancestor concept early in its gloss, rather than later. However this feature is not easy to generalize for other taxonomies.

5.3 Emergent Domain Concepts

We perform a case study to examine how ETF adds newly emerging concepts into the Wikipedia category hierarchy, focusing on the domains of crisis response and medicine (Table 4). We select a set of concepts, and eliminate them and all Wikipedia pages linking to them from consideration. To keep the evaluation process fair, we use as text input to our algorithm, the versions of the Wikipedia article pages on the concepts that existed when the concept was first added. Column 2 of Table 4 shows manually assigned parents

Table 4: Case Study: Performance of ETF on adding emergent domain concepts into the Wikipedia category taxonomy (first six rows). Quora Evaluation: example questions and comparison of manually assigned vs. predicted categories (last four rows).

New concept	Initial parents	Current manually assigned parents	Predicted parents by ETF
Tropical Storm Erika (2015)	Current events from August 2015	2015 Atlantic hurricane season, 2015 in the Caribbean, Natural disasters in Dominica, Hurricanes in Desirade, Hurricanes in Dominica, Hurricanes in Florida, Hurricanes in Guadeloupe, Hurricanes in Haiti, Hurricanes in the Bahamas, Hurricanes in Puerto Rico, Hurricanes in the Dominican Republic, Hurricanes in the Leeward Islands, Atlantic tropical storms	**Hurricanes in Florida**, *Natural disasters in Florida, Tropical cyclones,* Articles which contain graphical timelines, *Atlantic hurricanes*, **Hurricanes in Guadeloupe, Hurricanes in Dominica**, Tropical cyclone seasons, *Typhoons*, **2015 Atlantic hurricane season**, Tropical cyclones by strength, *Natural disasters in Guadeloupe*, Tropical cyclones by basin
Illapel Earthquake, Chile (2015)	Current events, 2015 earthquakes	2015 earthquakes, 2015 tsunamis, 2015 in Chile, Megathrust earthquakes in Chile, Tsunamis in Chile, September 2015 events	**2015 in Chile**, *Earthquakes in Chile*, **Tsunamis in Chile, 2015 tsunamis, 2015 earthquakes**, **Megathrust earthquakes in Chile**
Shootings at Parliament Hill, Ottawa (2014)	Current events, 39th Canadian Parliament	Attacks in 2014, 21st century in Ottawa, 41st Canadian Parliament, Terrorist incidents in Canada in 2014, Attacks on legislatures, Crime in Ontario, Deaths by firearm in Canada, October 2014 events, Parliament of Canada, Spree shootings in Canada, Political controversies in Canada, ISIL terrorist incidents in Canada, 2014 in Ontario	**2014 in Ontario**, Military history of Ontario, Ottawa, **Parliament of Canada**, Monuments and memorials in Ottawa, History of Ottawa, *Terrorist incidents in 2014*, **Crime in Ontario**, **Attacks in 2014**, Religion in Canada, **Political controversies in Canada, 21st century in Ottawa**, *Terrorist incidents in Canada*, **Spree shootings in Canada**
Avian Influenza	Current events, Influenza	Animal virology, Bird diseases, Avian influenza, Poultry diseases, Agricultural health and safety	*Subtypes of Influenza A virus*, **Bird diseases**, *Animal diseases*, **Poultry diseases**, *Viral diseases*
Influenza A virus subtype H7N9	Subtypes of Influenza A virus	Subtypes of Influenza A virus, 2013 health disasters, Health disasters in China, 2013 disasters in China	**2013 health disasters, Health disasters in China, 2013 disasters in China**, *Bird diseases*
Swine influenza	Medicine stubs, Pandemics	Animal virology, Health disasters, Swine diseases, Influenza, Pandemics	*2009 flu pandemic, Influenza A virus subtype H1N1*, **Influenza**, *Influenza pandemics*, **Health disasters**
Does overworking help Japan's economy?	–	Applied sciences, Social sciences, Economy of Japan, Processes, Economy of Asia, Economies by country, Economics, Japan	*Social impact*, **Economy of Japan**, **Economies by country**, Retailing by country, **Economy of Asia**, *Economies by region, Economywide country studies*
Why do lightnings have a branch structure?	–	Space plasmas, Atmospheric electricity, Lightning, Storm, Electrical phenomena, Weather hazards, Electric arcs, Electrical breakdown	*Meteorological phenomena*, Electric power transmission, Electric power, *Physical phenomena*, **Weather hazards, Electrical breakdown, Storm**
How true is the belief that natural things are good?	–	Main topic classifications, Science, Science technology engineering and mathematics, Nature, Physical universe	**Nature**, *Philosophical theories*, Biological interactions, **Physical universe**, History of science by discipline
What did Han Xin do wrong to be killed by Liu Bang?	–	China, Asian royal families, Chinese-speaking countries and territories, Han dynasty, History of Ancient China, History of China, Dynasties in Chinese history, Iron Age Asia	**Dynasties in Chinese history**, *Han dynasty people*, Qin Dynasty, **Asian royal families**, History of Asia by country, **History of Ancient China**, *People by Imperial Chinese dynasty, Histories of cities in China*

for the original versions of the new concept pages. Column 3 shows the manually allocated categories on the present-day Wikipedia versions of the new concept pages, and column 4 shows the top ranked parents predicted by our approach. In column 4, we show in bold the predicted parents that match with those in column 3, and italicize the parents that are good predictions but are not in column 3. For both domains, most predicted parents are quite good without being overly general, and overlap with many of the manually assigned parents. For the crisis events (rows 1, 2, 3), ETF correctly identifies their year of occurrence and a good number of affected areas, with just a preliminary amount of text input. With respect to the medical domain, our method can seemingly discriminate between *Avian Influenza* and *Swine Influenza* (rows 4, 5, 6). We are also able to propose accurate categories for new concepts that have not been manually assigned: *Subtypes of Influenza A virus* for *Avian influenza*, and *2009 flu pandemic* for *Swine Influenza*.

Overall, the parents predicted by ETF for all the emergent concepts are better than the human-assigned categories to them at that point in time. The results indicate that our algorithm can accurately organize new concepts across varied domains.

5.4 Quora Q&A Categorization

Finally, as a use case for ETF, we consider the problem of mapping questions and answers (Q&A) from Quora to appropriate Wikipedia categories (see last 4 rows of Table 4). These questions and answers have a different style of writing than the more definitional style of Wikipedia articles. Moreover, a single question and answer can span many topics. One feature of Quora is that many of the existing tags assigned to questions directly map to existing categories and entities from Wikipedia. We selected 384 questions from a diverse selection of categories, and processed the first paragraph of the question and top answer using our framework, treating this text as a definition for a new concept. On average each question had been tagged with 2 concepts from Wikipedia. Some of these concepts were entities, and some of them were categories. For the ground truth, we consider the union of the tagged categories and parent categories of the tagged entities. Overall, this resulted in an average of 8 parents per Q&A. ETF achieved an NDCG of 0.445 and F1 score of 0.523, and was, similar to previous results, an improvement (NDCG increase of 5.5%, and F1 increase of 8.7%) over the other baselines. The scores on this test were lower across all aproaches, compared to the Wikipedia-based case study. We believe the primary reasons for this drop are that the ground truth is much sparser, and also that this task is more difficult than that of Section 5.1, as there is more concept mixing in the Quora Q&A compared to the focused writing on Wikipedia. However, we believe that ETF could be adapted to automate such a tagging system in the future, reducing the annotation effort.

6 CONCLUSION AND FUTURE WORK

In this work, we propose a solution to the problem of automated taxonomy enrichment, where we insert new concepts at appropriate positions into an existing taxonomy. Our proposed approach ETF learns a high-dimensional vector embedding via a generated context of terms, for each existing and new concept. We then predict the potential parents of the new concepts from the ancestors of their nearest neighbors, by ranking them based on semantic and graph features. We evaluated ETF on the large knowledge bases of Wikipedia and WordNet, and could outperform other baselines.

ETF has the potential to be applicable under variations in text sources (e.g. short, informal social media text) and types of taxonomies (e.g. enhancing taxonomies belonging to specific domains). It allows for easy parallelization and can be distributed for scalability. All features used by ETF can be computed in a few seconds except the random walk betweenness centrality feature, though we leave a detailed analysis and discussion for future work.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under the grant EAR-1520870. Computational support was provided by the Ohio Supercomputer Center. All content represents the opinion of the authors, which is not necessarily endorsed by their sponsors.

REFERENCES

[1] L. Bentivogli, A. Bocco, and E. Pianta. 2004. ArchiWordnet: integrating Wordnet with domain-specific knowledge. In *Global Wordnet Conference.*

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD.*

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS.*

[4] P. Buitelaar and B. Sacaleanu. 2002. Extending synsets with medical terms. In *International WordNet Conference.*

[5] C.J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report.

[6] C. Fellbaum. 1998. WordNet: An Electronic Lexical Database. (1998).

[7] C. Fellbaum, U. Hahn, and B. Smith. 2006. Towards new information resources for public health. *Journal of Biomedical Informatics* (2006).

[8] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu. 2014. Learning semantic hierarchies via word embeddings. In *ACL Vol 1 Long Papers.*

[9] M.A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *ACL COLING.*

[10] http://wiki.dbpedia.org/dbpedia-version 2015-10. 2015. DBPedia version. (2015).

[11] Z. Huang, W. Xu, and K. Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv:1508.01991* (2015).

[12] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene. 2013. Unsupervised graph-based topic labelling using dbpedia. In *ACM WSDM.*

[13] T. Joachims. 2006. Training linear SVMs in linear time. In *ACM SIGKDD.*

[14] D. Jurgens and M.T. Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the WordNet taxonomy for novel terms.. In *NAACL.*

[15] D. Jurgens and M.T. Pilehvar. 2016. SemEval-2016 Task 14: Semantic Taxonomy Enrichment.. In *SemEval@ NAACL-HLT.*

[16] A. Karpathy and L. Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE CVPR.*

[17] L. Katz. 1953. A new status index derived from sociometric analysis. In *Psychometrika.*

[18] P. Lahoti, P.K. Nicholson, and B. Taneva. 2017. Efficient Set Intersection Counting Algorithm for Text Similarity Measures. In *ALENEX.*

[19] Q. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *ICML.*

[20] B. Li, J. Liu, C. Lin, I. King, and M. Lyu. 2013. A hierarchical entity-based approach to structuralize user generated content in social media. In *EMNLP.*

[21] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion.. In *AAAI.*

[22] T. Mai, B. Shi, P.K. Nicholson, D. Ajwani, and A. Sala. 2017. Scalable Disambiguation System Capturing Individualities of Mentions. In *LDK.* Springer.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representation of words and phrases and their compositionality. In *NIPS.*

[24] D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* (2007).

[25] R. Navigli, P. Velardi, and S. Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI.*

[26] M.E.J. Newman. 2005. A measure of betweenness centrality based on random walks. *Social networks* (2005).

[27] B. Perozzi, R. Al-Rfou, and S. Skiena. 2014. Deepwalk: Online learning of social representations. In *ACM SIGKDD.*

[28] M. Poprat, E. Beisswanger, and U. Hahn. 2008. Building a BioWordNet by using WordNet's data formats and WordNet's software infrastructure. In *Software engineering, testing, and quality assurance for natural language processing.*

[29] R. Řehůřek and P. Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC Workshop on New Challenges for NLP Frameworks.*

[30] M. Schlichtkrull and H.M. Alonso. 2016. MSejrKu at SemEval-2016 Task 14: Taxonomy Enrichment by Evidence Ranking. In *SemEval.*

[31] R. Snow, D. Jurafsky, and A.Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS.*

[32] F. Suchanek, G. Kasneci, and G. Weikum. 2007. Yago: a core of semantic knowledge. In *ACM WWW.*

[33] A. Sumida and K. Torisawa. 2008. Hacking Wikipedia for Hyponymy Relation Acquisition.. In *IJCNLP.*

[34] J. Sun, D. Ajwani, P.K. Nicholson, A. Sala, and S. Parthasarathy. 2017. Breaking Cycles In Noisy Hierarchies. In *ACM WebSci.*

[35] O. Suominen and E. Hyvönen. 2012. Improving the quality of SKOS vocabularies with Skosify. In *Knowledge Engineering and Knowledge Management.*

[36] L. Tan, R. Gupta, and J. van Genabith. 2015. USAAR-WLV: Hypernym Generation with Deep Neural Nets.. In *SemEval@ NAACL-HLT.*

[37] A. Toral and M. Monachini. 2008. Named entity wordnet. In *LREC.*

[38] L. Tuan, Y. Tay, S. Hui, and S. Ng. 2016. Learning Term Embeddings for Taxonomic Relation Identification Using Dynamic Weighting Neural Network. In *EMNLP.*

[39] N. Vedula and S. Parthasarathy. 2017. Emotional and Linguistic Cues of Depression in Social Media. In *ACM Digital Health (DH).*

[40] N. Vedula, W. Sun, H. Lee, H. Gupta, M. Ogihara, J. Johnson, G. Ren, and S. Parthasarathy. 2017. Multimodal Content Analysis for Effective Advertisements on YouTube. In *IEEE ICDM.*

[41] P. Vossen. 2001. Extending, trimming and fusing WordNet for technical documents. ACL.

[42] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han. 2013. A phrase mining framework for recursive construction of a topical hierarchy. In *ACM SIGKDD.*

[43] Z. Wang, J. Zhang, J. Feng, and Z. Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes.. In *AAAI.*

[44] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade.*

[45] D. Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *NAACL-HLT.*

[46] Z. Wu and M. Palmer. 1994. Verbs semantics and lexical selection. In *ACL.*

[47] I. Yamada, J. Oh, C. Hashimoto, K. Torisawa, Jun'ichi K., S. De Saeger, and T. Kawada. 2011. Extending WordNet with Hypernyms and Siblings Acquired from Wikipedia.. In *IJCNLP.*

[48] I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Saeger, F. Bond, and A. Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *EMNLP.*

[49] H. Yang and J. Callan. 2009. A metric-based framework for automatic taxonomy induction. In *ACL.*

[50] Z. Yang, W.W. Cohen, and R. Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *arXiv:1603.08861* (2016).

[51] Z. Yu, H. Wang, X. Lin, and M. Wang. 2015. Learning Term Embeddings for Hypernymy Identification.. In *IJCAI.*