Geometric Estimation of Specificity within Embedding Spaces

Negar Arabzadeh
Ryerson University
Toronto, Ontario
Narabzad@ryerson.ca

Fattane Zarrinkalam
Ryerson University
Toronto, Ontario
fzarrinkalam@ryerson.ca

Jelena Jovanovic
University of Belgrade
Serbia
jelena.jovanovic@fon.bg.ac.rs

Ebrahim Bagheri
Ryerson University
Toronto, Ontario
bagheri@ryerson.ca

ABSTRACT

Specificity is the level of detail at which a given term is represented. Existing approaches to estimating term specificity are primarily dependent on corpus-level frequency statistics. In this work, we explore how neural embeddings can be used to define corpus-independent specificity metrics. Particularly, we propose to measure *term specificity* based on the distribution of terms in the neighborhood of the given term in the embedding space. The intuition is that a term that is surrounded by other terms in the embedding space is more likely to be *specific* while a term surrounded by less closely related terms is more likely to be *generic*. On this basis, we leverage geometric properties between embedded terms to define three groups of metrics: (1) neighborhood-based, (2) graph-based and (3) cluster-based metrics. Moreover, we employ learning-to-rank techniques to estimate term specificity in a supervised approach by employing the three proposed groups of metrics. We curate and publicly share a test collection of term specificity measurements defined based on Wikipedia’s category hierarchy. We report on our experiments through metric performance comparison, ablation study and comparison against the state-of-the-art baselines.

ACM Reference Format:
Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, and Ebrahim Bagheri. 2019. Geometric Estimation of Specificity within Embedding Spaces. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM ’19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358152

1 INTRODUCTION

Specificity, often defined as inversely related to ambiguity, has traditionally been estimated using corpus-specific frequency statistics [2]. More recently, the information retrieval community has embarked on exploring the impact of neural embeddings in different applications such as query expansion, query classification, and ranking, just to name a few [5, 6]. Neural embeddings maintain interesting *geometric properties* where the direction and magnitude of relationship between the vector representation of terms derived from embeddings are meaningful. In this paper, we explore how the geometric properties of neural embeddings can be exploited to define individual and collective *specificity metrics.* While corpus-level term frequency information is not explicitly maintained in the neural embeddings of terms, inter-term associations can be estimated based on how the vector representations of terms are distributed within the embedding space. This suggests that by considering the associations between term vectors in the neural embedding space, we can go beyond frequency-based metrics and derive other measures of specificity. More specifically, we propose to measure *term specificity* based on the distribution of terms that form the neighborhood of a term of interest in the embedding space. Our metrics are based on the intuition that a term that is closely surrounded by other terms in the embedding space is more likely to be *specific* while a term with less nearby terms is more likely to be *generic.* On this basis, we conceptualize the embedding space surrounding a term by defining an *ego network* where the term of interest forms the ego and is contextualized within a set of alter nodes, which are other terms that are closely positioned around the term of interest in the embedding space. In the context of such an ego network, we define three groups of metrics: (1) *neighborhood based*, which are based on the idea that a specific term is likely to be associated with a large number of terms in its neighborhood. (2) *graph-based metrics*, which consider the structure of the ego network to estimate the specificity of the ego node, and, (3) *cluster-based metrics*, which consider the term clusters around a term as potential indicators for the specificity of the term.

In order to evaluate the proposed specificity metrics, we introduce and publicly share a test collection, which is a structured collection of terms with associated human-defined specificity values. The test collection has been derived from the Wikipedia category hierarchy with the understanding that categories higher up in the hierarchy are more generic, while those further down in the hierarchy are more specific. Specifically, we have extracted sequences of categories from the Wikipedia’s category hierarchy, where each sequence consists of semantically related categories with progressively higher degree of specificity, reflecting the change in specificity from the higher towards the lower levels in the hierarchy. These sequences are used to measure the performance of the proposed neural embedding-based specificity metrics.

In summary, we provide the following contributions in this paper: (1) We formally introduce the task of predicting corpus-independent

CIKM ’19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358152

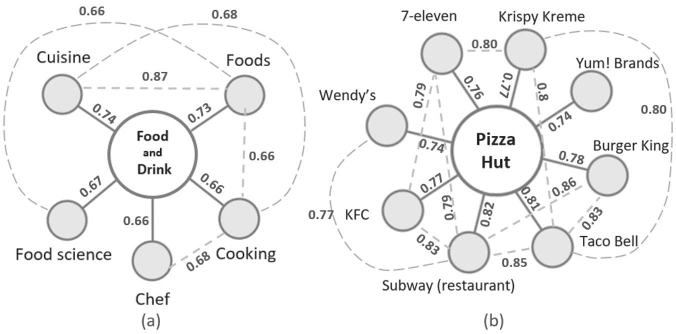

Figure 1: Ego-networks for: a) Food and Drink, b) Pizza Hut

term specificity based on a collection of neural embeddings; (2) We propose a host of unsupervised and supervised metrics for predicting specificity based on embeddings; and (3) We present a test collection consisting of term specificity measurements defined in relation to Wikipedia categories. We publicly share our curated test collection for future replication studies[1].

2 SPECIFICITY METRICS

Our work focuses on how vector representations of terms within a given embedding space could be used to define appropriate metrics for estimating term specificity. Our intuition is that the neighborhood of a term in a given embedding space can be used to derive indicators of the term’s specificity. Based on the fact that, in an embedding space, two semantically related terms have similar embedding vectors, we select the local neighborhood surrounding an embedding vector of term t_i, denoted as v_{t_i}, by retrieving a set of highly similar terms to t_i. More specifically, let $\mu(t_i)$ be the degree of similarity of the most similar term to t_i in the embedding space. We first calculate the cosine similarity between the embedding vector of t_i and other terms’ embedding vectors in the embedding space. Terms with a similarity higher than $\varepsilon * \mu(t_i)$, are selected as the ε-neighborhood of term t_i, denoted as $N_\varepsilon(t_i)$. Next, in order to show inter-term associations in the neighborhood of t_i, we formalize the notion of an *ego network*, which is based on term similarities within the neural embedding space, as follows:

Definition 2.1. (Ego Network) An ego network for term t_i, denoted as $\xi(t_i) = (\mathbb{V}, \mathbb{E}, g)$, is a weighted undirected graph where $\mathbb{V} = \{t_i\} \cup N_\varepsilon(t_i)$, and $\mathbb{E} = \{e_{t_i,t_j} : \forall t_i, t_j \in \mathbb{V}\}$. The function $g : \mathbb{E} \rightarrow [0, 1]$ is the cosine similarity between the embedding vectors of two incident terms of an edge e_{t_i,t_j}, i.e., v_{t_i} and v_{t_j}. We refine $\xi(t_i)$ by pruning any edge with a weight below $\varepsilon * \mu(t_i)$.

Simply put, we build an ego network for term t_i such that t_i is the ego node and is connected directly to other terms only if the degree of similarity between the ego and its neighbors is above a given threshold. For instance, assuming ‘Pizza Hut’ is the ego and $\varepsilon = 0.9$, given the fact that ‘Subway (restaurant)’ is the most similar term to the ego with a similarity of 0.82, the immediate neighbors of the ego node will consist of all the terms in the embedding space that have a similarity above 0.738 to ‘Pizza Hut’. Figure 1 shows the ego networks for the specific term ‘Pizza Hut’ and the generic term ‘Food and Drink’. As shown in the figure, for example, the immediate neighbors of ‘Pizza Hut’ include terms such as ‘Subway (restaurant)’, ‘KFC’, ‘7-Eleven’, ‘Burger King’, among others.

Table 1: The set of specificity metrics in our work.

Neighborhood-based Metrics	
neighborhood Size (NS)	Number of terms in t_i neighborhood
Weighted Degree Centrality (WDC)	Average weight of edges connected to t_i
Median Absolute Deviation (MAD)	Median Absolute Deviation (MAD) of weight of edges connected to t_i
neighborhood Variance (NV)	Variance of weight of t_i edges
Most Similar neighbor (MSN)	The maximum weight of edges connected to t_i
Graph-based Metrics	
Edge Count (EC)	Number of edges in the network
Edge Weight Sum (EWS)	The sum of edge weights in the network
Edge Weight Avg_ego (EWAe)	Average of edge weights in the network
Edge Weight Max_ego (EWXe)	Minimum edge weight in the network
PageRank (PR)	PageRank of t_i in the network
Cluster-based Metrics	
Clusters Elements Variance (CEV)	Variance of number of elements in extracted clusters of the network
Edge Weight Avg_centroid (EWAc)	Average edge weight in centroid network of t_i
Edge Weight Min_centroid (EWNc)	Minimum edge weight in centroid network of t_i
Edge Weight Max_centroid (EWXc)	Maximum edge weight in centroid network of t_i

Based on the developed ego network for a given term, we propose to measure the specificity of the ego term. In particular, based on our intuition that the characteristics of the local neighborhood of a term are potential indicators of its specificity, we measure term specificity as a function of the structure of the term’s ego network. In the following, we first propose three categories of unsupervised specificity metrics, namely (1) neighborhood-based (ego-node) metrics; (2) graph-based (ego-network) metrics; and (3) cluster-based metrics. In Table 1, we summarize the metrics defined in each category. Then, we propose a supervised method that incorporates all these metrics, as features, in *learning to rank* for estimating the specificity of a term based on neural embeddings.

Neighborhood-based Metrics: Neighborhood based metrics only consider the connections between the ego node and its immediate neighbors. Our intuition is that as highly specific terms express precise semantics, they have a high likelihood of being surrounded, in the embedding space, by a higher number of specific terms compared to generic terms. For example, the specific term ‘Pizza Hut’ which refers to a fast food brand is highly similar to other terms referring to other fast food chains such as ‘KFC’ and ‘Burger King’. However, since a generic term, e.g., ‘Food and Drink’, is often related to many different terms with diverse senses, it would end up having weaker relationships with these diverse neighbors. In other words, generic terms are likely to be related to other generic terms that originate from various domains, and while the relation (i.e., semantic relatedness) between terms does exist, it is weaker than in the case of specific terms that are highly semantically related to one another. Therefore, by considering the strength of connection (edge weight) of a term with its immediate neighbors, it is possible to differentiate between specific and generic terms.

Graph-based Metrics: While neighborhood-based metrics only focus on the connections of neighborhood terms with the ego node, in graph-based metrics, we take all the connections in the ego-network into account. Our intuition for these metrics is that the denser an ego-network is, the more specific the ego term would

[1] https://github.com/WikipediaHierarchyPaths/WikiPedia_Hierarchy_Paths

be. In other words, not only a specific term is surrounded by a higher number of neighbors in the embedding space, but also its neighbors are highly similar to each other. For instance, the average edge weights of the ego-network of the generic term 'Food and Drink' (0.705) is less than for the specific term 'Pizza Hut' (0.795). In addition, as shown in Figure 1, the number of edges in 'Food and Drink' ego network is less than in the ego network for 'Pizza Hut'.

Cluster-based Metrics: These metrics are based on the idea that the characteristics of term clusters within the neighborhood of a given term are potential indicators of its specificity. Therefore, to extract the term clusters around t_i, we apply a clustering algorithm, such as K-means, to the embedding vectors of terms in its ε-neighborhood, i.e., $N_\varepsilon(t_i)$, which results in K clusters for t_i, i.e., $C^1_{t_i}, ..., C^K_{t_i}$. Then, we estimate the specificity of t_i by calculating the variance of the number of elements in the obtained clusters. As mentioned before, a generic term is more likely to be related to many terms from different domains. If we cluster the neighborhood terms of a term, we may expect that each cluster will be associated with one domain. Low variance of the number of elements in the clusters shows that there is no dominant cluster in the neighborhood of the term. Consequently, it is probably a more generic term. The association between the clusters can also be considered an indicator of specificity. Each cluster is defined with its *centroid* c, which is a vector in the embedding space that indicates the center of the cluster. Therefore, for term t_i, given its clusters, i.e., $C^1_{t_i}, ..., C^K_{t_i}$, we define its centroid network, denoted as $\zeta(t_i)$, as follows:

Definition 2.2. (Centroid Network) A centroid network for term t_i, denoted as $\zeta(t_i) = (\mathbb{V}, \mathbb{E}, g)$, is a weighted undirected graph in which $\mathbb{V}$ includes the centroid points of the term's clusters $C^1_{t_i}, ..., C^K_{t_i}$, and $\mathbb{E} = \{e_{c_i,c_j} : \forall c_i, c_j \in \mathbb{V}\}$. The weight function $g : \mathbb{E} \rightarrow [0, 1]$ is the cosine similarity between the vectors of two centroid points of an edge e_{c_i,c_j}, i.e., v_{c_i} and v_{c_j}.

The idea is that the more the term clusters of a given term are similar, the more specific the terms would be. Therefore, edge weights in the centroid network play crucial role in estimating specificity because they show how clusters are distributed in the embedding space. We define three metrics by aggregating edge weights in the centroid network using min, max and average functions.

Supervised method: We additionally apply a supervised strategy to collectively incorporate all the unsupervised specificity metrics to predict the specificity of a term. We take advantage of the learning to rank strategy to predict the specificity of terms by representing each term by a vector of features. In our model, features are the same as the specificity metrics in Table 1.

3 TEST COLLECTION

We introduce a test collection to evaluate the proposed measures of term specificity. In order to avoid the biases associated with manually curated gold standard datasets, we based our test collection on the Wikipedia category hierarchy, which formally organizes knowledge in degrees of specificity. The most generic category of the hierarchy sits at the top most level and the most specific categories are located at the leaves of the hierarchy. Since the level of each node in the hierarchy is an appropriate indicator of the node's specificity with regards to its parent and child nodes, it can be used as ground truth to evaluate specificity metrics. We used Wikipedia dumps dated April 2016. This dataset consists of 1,411,022 categories with 2,830,740 subcategory relations between them. Kapanipathi et al. [3] have empirically found that while hierarchical, the Wikipedia category can potentially include cyclic references between categories. Therefore, to transform the Wikipedia category structure into a strict hierarchy, we adopt the approach proposed by Kapanipathi et al. [3]. The outcome of this process is a hierarchy with a height of 26 and 1,016,584 categories with 1,486,019 links.

We consider the level of each category, i.e., its shortest path to the root, as an indicator for the specificity of that category. As such and in order to evaluate specificity metrics, we have randomly sampled 713 unique paths each with a length of 5, which form our test collection. Based on the paths included in the test collection and given a set of categories, the objective of an effective specificity metric would be to produce the correct ordering that exists in the test collection. It is then possible to evaluate the performance of each specificity metric using rank correlation measures to determine the relationship between the actual order of categories and the ranking based on the specificity metrics. Our test collection consists of two types of paths; *narrow-ranged paths* and *wide-ranged paths*. The narrow-ranged paths consist of categories that are observed immediately one after the other in the Wikipedia category hierarchy. In contrast, in the wide-ranged paths subsequent categories in each path are guaranteed to have a distance of at least one hop from each other. The reason for these two types of paths is that given the immediacy of category neighborhood in narrow-ranged paths, it would be much harder to correctly estimate the categories relative specificity compared to the wide-ranged paths. As such, the performance of the specificity metrics over these two types of paths allows us to compare the sensitivity of the specificity metrics.

4 EXPERIMENTS

Given our test collection is based on Wikipedia, we needed an embedding model based on Wikipedia content and its categories. We adopt the pre-trained Hierarchical Category Embedding (HCE) model [4]. Moreover, ε in ε-neighborhood of each term is set based on five-fold cross validation optimized for Kendall Tau.

We present the results of the experiments that compare the proposed unsupervised specificity metrics based on Kendall Tau rank correlation in Table2. The Top-3 performing metrics in ranking both narrow-ranged and wide-ranged paths are EWAe, WDC, and EWNc. We have observed that graph-based metrics in general show better performance compared to their neighborhood-based counterparts. A potential explanation may be that while the number of neighbors computed in the neighborhood-based metrics is intuitively an indicator for specificity, the connections between neighboring nodes captured in the graph-based metrics can reinforce and strengthen specificity estimation. This hypothesis is boosted when comparing EWAe and WDC. WDC that overlooks the connections between neighboring nodes performs worse than EWAe that considers such connections.

4.1 Learning Specificity

To apply learning to rank, we used RankLib and exploited three well-known methods, namely RankBoost, RandomForest, and Coordinate Ascent by optimizing ndcg@5. Table 3 reports the performance of the models obtained by measuring Kendall Tau between the actual order of categories observed in the test collection and

Table 2: Performance of the metrics on Kendall Tau. All values are statistically significant at alpha=0.05 (paired t-test).

	Neighborhood-based Metrics					Graph-based Metrics					Cluster-based Metrics			
	NS	WDC	MAD	NV	MS	EC	EWS	EWAe	EWXe	PR	CEV	EWAc	EWNc	EWXc
Narrow-ranged paths	0.091	**0.269**	0.174	0.097	0.166	0.187	0.092	**0.310**	0.101	0.054	0.051	0.259	**0.274**	0.222
Wide-ranged paths	0.072	**0.46**	0.348	0.145	0.311	0.273	0.210	**0.510**	0.173	0.206	0.062	0.456	**0.481**	0.392

Table 3: Comparison of learning-to-rank models in terms of Kendall Tau to top-3 best performing metrics from Table 2.

	Supervised Methods			Top-3 Unsupervised Metrics		
	RankBoost	Random Forest	Coordinate Ascent	WDC	EWAe	EWNc
Narrow-ranged Paths	0.314	**0.314**	0.311	0.269	0.310	0.274
Wide-ranged Paths	0.539	**0.595**	0.561	0.46	0.510	0.481

Table 4: Performance of Random Forest after incrementally adding features on narrow-ranged paths. Base model includes 4 most important features: EWNc, CEV, EWXe and MSN. * indicates statistical significance on a paired t-test p-value<0.05.

	Base Model	+EWS	+MAD	+EWAc	+NS	+EWXc	+EWAe	+WDC	+EC	+NV	+PR
	0.257	0.297	0.313	0.294	0.302	0.297	0.308	0.312	0.317	0.315	0.314
Δ		+15.56%*	+5.38%*	-6.07%	+2.72%	-1.65	+3.7%*	+1.29%	+1.6%	-0.63%	-0.31%

Table 5: Similar to Table 4 but for wide-ranged paths. Base model includes: EWNc, EWAc, EWXe and CEV.

	Base Model	+EWXc	+EWS	+EWAe	+NS	+MSN	+NV	+WDC	+EC	+MAD	+PR
	0.48	0.491	0.493	0.527	0.52	0.594	0.581	0.6	0.585	0.601	0.595
Δ		+0.82%	+0.4%	+6.89%*	-1.32%	+7.4%*	-1.3%	+3.2%*	-2.5%	+2.73%	-0.99%

Table 6: Comparison with Frequency-based Baselines.

	Baselines		Our method (best variation)	
	IDF_{max}	SCS	Supervised	Unsupervised
Narrow-ranged	0.294*	0.086*	0.314*	0.31*
Wide-ranged	0.315*	0.144*	0.595*	0.51*

the list of categories ranked based on the output of each learning to rank model. Note that the results reported in Table 3 are calculated by applying five-fold cross validation, for each learning-to-rank method. As seen in the table, the application of learning-to-rank methods that incorporate all of the proposed specificity metrics in a single model outperforms the Top-3 specificity metrics when used in isolation in both narrow-ranged and wide-ranged paths. To analyze the relative effectiveness of each feature, we rank all features based on their feature frequency in the Random Forest model. We start with a base model consisting of 4 most important features and proceed to extend this model by incrementally introducing additional features, based on their frequency. In Tables 4 and 5 the results of this ablation study are reported in terms of Kendall Tau rank correlation. Three out of four most important features for both narrow-ranged and wide-ranged paths are EWNc, EWXe and CEV. It is interesting to note that EWNc is also among the top-3 performing metrics based on the results of evaluating each metric separately reported in Table 2. In Table 4, for narrow-ranged paths when Edge Weight Sum (EWS) is added as a feature to the base model, we observe a significant improvement of about 15%. Two other significant improvements are also observed by adding MAD and EWAe. Adding the rest of features does not lead to significant changes in performance. For wide-ranged paths, the significant improvements occur when adding EWAe, MSN and WDC.

4.2 Comparison with Baseline Metrics

We also compare against two well-known frequency-based specificity metrics, i.e., Max IDF and Simplified Clarity Score (SCS) [1]. In order to apply these metrics we consider Wikipedia as a collection of documents and compute Max IDF and Simplified Clarity Score (SCS) metrics. In Table 6, the results of the two frequency-based baselines are compared to the best variations of our supervised and unsupervised models. As shown, both of our methods outperform the baselines despite the fact that the baseline methods have access to corpus-specific frequency information, whereas our methods do not and are solely based on pre-trained neural embeddings.

5 CONCLUDING REMARKS

We have introduced three types of unsupervised metrics as well as a supervised learn to rank strategy for measuring term specificity based on neural embeddings. We have also curated and publicly shared a test collection to serve as the gold standard. Our key findings include: (1) pre-trained, corpus-independent neural embedding representations of terms enable accurate estimates of term specificity, (2) graph-based specificity metrics that consider term neighborhood as well as term associations have the best performance, (3) the three proposed types of metrics have synergistic impact on specificity estimation, as demonstrated through the ablation study, and (4) the corpus-independent metrics perform better than traditional frequency-based corpus-dependent specificity metrics.

REFERENCES

[1] Ben He and Iadh Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors.. In *String Processing and Information Retrieval, 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004, Proceedings*. 43–54.

[2] Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 60, 5 (2004), 493–502.

[3] P. Kapanipathi, P. Jain, C. Venkatramani, and A. P. Sheth. 2014. User Interests Identification on Twitter Using a Hierarchical Knowledge Base. In *ESWC*. 99–113.

[4] Y. Li, R. Zheng, T. Tian, Z. Hu, R. Iyer, and K. P. Sycara. 2016. Joint Embedding of Hierarchical Categories and Entities for Concept Categorization and Dataless Classification. In *COLING*. 2678–2688.

[5] Bhaskar Mitra and Nick Craswell. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends in Information Retrieval* 13, 1 (2018), 1–126.

[6] H. Zamani, W. B. Croft, and J. S. Culpepper. 2018. Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *SIGIR*. 105–114.