Navigating the Semantic Horizon using Relative Neighborhood Graphs

Amaru Cuba Gyllensten and Magnus Sahlgren
Gavagai
Bondegatan 21
116 33 Stockholm
Sweden
{amaru|mange}@gavagai.se

Abstract

This paper introduces a novel way to navigate neighborhoods in distributional semantic models. The approach is based on *relative neighborhood graphs*, which uncover the topological structure of local neighborhoods in semantic space. This has the potential to overcome both the problem with selecting a proper k in k-NN search, and the problem that a ranked list of neighbors may conflate several different senses. We provide both qualitative and quantitative results that support the viability of the proposed method.

1 Introduction

Nearest neighbor search is a fundamental operation in data mining, in which we are interested in finding the closest points to some given reference point. Formally, if we have a reference point r and a set of other points P in a metric space M with some distance function d, the nearest neighbor search task is to find the point $p \in P$ that minimizes $d(p, r)$. In k-Nearest Neighbor search (k-NN), we want to find the k closest points to some given reference point. Nearest neighbor search is a well-studied task, and in particular the complexity of the task (a linear search has a running time of $\mathcal{O}(Ni)$ where N is the cardinality of P and i the complexity of the distance function d) has generated a lot of research; suggestions for reducing the complexity of linear nearest neighbor searches include using various types of space partitioning techniques like k-d trees (Bentley, 1975), or various techniques for doing *approximate* nearest neighbor search (Arya et al., 1998), of which one of the most well-known is locality-sensitive hashing (Indyk and Motwani, 1998).

The problem we are concerned with in this paper is not the complexity of nearest neighbor search, but the question of *how to identify the internal structure of neighborhoods defined by the nearest neighbors*. The problem with a normal k-NN is that the result — a sorted list of the k nearest neighbors — does not say anything about the internal structure of the neighborhood. It is quite possible for two neighborhoods with widely different internal structures to produce identical k-NN results. In the context of *Distributional Semantic Models* (DSMs), which collect and represent co-occurrence statistics in high-dimensional vector spaces, such structural differences may carry significant semantic information, e.g. about the different senses of terms. We argue that the inability of standard k-NN to account for structural properties has been misinterpreted as a shortcoming of the distributional representation (Erk and Padó, 2010).

We will demonstrate in this paper that this is *not* a shortcoming of the distributional representation, but of the *mode of querying* the DSM. We argue that information about the different usages (i.e. senses) of a term is encoded in the structural properties of the nearest neighborhoods, and we propose the use of *relative neighborhood graphs* for identifying these structural properties. Relative neighborhood graphs may also be used for finding a relevant k for a given reference point, which we refer to as the *horizon* with respect to the reference point.

2 Distributional Semantics and Nearest Neighbor Search

Collecting and comparing co-occurrence statistics for terms in language has become a standard approach for computational semantics, and is now commonly referred to as *distributional semantics*. There are many different types of models that can be used for this purpose, but their common objective is to represent terms as vectors that record (some function of) their distri-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2451–2460,
Lisbon, Portugal, 17-21 September 2015.

butional properties. The standard approach for generating such vectors is to collect distributional statistics in a *co-occurrence matrix* that records co-occurrence counts between terms and contexts. The co-occurrence matrix is then subject to various types of transformations, ranging from the application of simple frequency filters or association measures to matrix factorization or regression models. The resulting representations are referred to as *distributional vectors* (or *word embeddings*), which are used to compute similarity between terms.

Given a similarity — or distance — measure on such distributional vectors, we can perform a nearest neighbor search. This is a particularly important operation in distributional semantics, since it answers the question "which other terms are similar to this one?" and this is a central question in semantics; lexica and thesauri are built with the main purpose of answering this question. Consequently, nearest neighbor search in a DSM could be seen as a compilation step in a distributional lexicon.

The result of a nearest neighbor search in a DSM is often presented as a list of (the top k) neighbors, sorted by descending similarity with the target term. Table 1 illustrates typical sorted nearest neighbor lists produced with three different DSMs: a standard model based on Pointwise Mutual Information (PMI)[1] that has been reduced to 2,000 dimensions by applying a Gaussian random projection; GloVe, which uses regression to find distributional vectors such that their dot product approximates their log probability of co-occurring (Pennington et al., 2014); and the Skipgram model, which uses stochastic gradient descent and hierarchical softmax combined with negative sampling and subsampling to find distributional vectors that maximize the probability of observed co-occurrence events (Mikolov et al., 2013). We refer to the respective papers for details regarding the various models. The similarity measure used is the cosine similarity: $s(a,b) = \frac{a \cdot b}{\|a\|\|b\|}$.

Table 1 lists the 10 nearest neighbors to *suit* in these three different DSMs using the entire Wikipedia as data. As can be expected, there are both similarities and dissimilarities between these neighborhoods; "suits" and "lawsuit" occur among the 10 nearest neighbors to "suit" in all three models, whereas other terms are specific for one particular model. What is common between the three models is that they all feature neighbors that represent two different usages of "suit": the *law*-sense ("lawsuit") and the *clothes*-sense ("dress", "wearing", "doublebreasted").[2] However, these distinction are not discernible by merely looking at the list of nearest neighbors; the only information it provides is the ranking of the nearest neighbors in descending order of similarity.

Table 1: Sorted list of the nearest neighbors to "suit" in three different distributional models.

PMI	**GloVe**	**Skipgram**
suits	suits	suits
dress	lawsuit	lawsuit
jacket	filed	countersuit
wearing	case	classaction
hat	wearing	doublebreasted
trousers	laiming	skintight
costume	lawsuits	necktie
shirt	alleging	wetsuit
pants	alleges	crossbone
lawsuit	classaction	lawsuits

It has been argued that DSMs that represent terms by a single vector cannot adequately handle polysemy, since they conflate several different usage patterns in one and the same vector (Véronis, 2004; Erk and Padó, 2010). Examples like the one above is often cited as evidence. We argue that this critique is unfounded and misinformed, and that it is *the mode of querying* the DSM that can be susceptible to problems with polysemy. As the above example demonstrates, querying DSMs by k-NN conflates different usages of terms. The reason for this seems quite obvious: simply ranking the nearest neighbors by similarity (or distance) ignores any local structures of the neighborhood. If "suit" has as neighbors both "dress" and "lawsuit", which represent two distinct types of usages of "suit", there will be a *structural* distinction in the neighborhood of "suit" between these different neighbors, since they will be mutually unrelated (i.e. there is a similarity between "suit" and

[1] For observations a and b, PMI$(a,b) = \log \frac{p(a,b)}{p(a)p(b)}$. The probabilities are often replaced in DSMs by co-occurrence counts of a and b and their respective frequency counts.

[2] The Skipgram model also features a *manga*-related sense of "suit" in the neighbor "crossbone," which refers to the mange series "Mobile Suit Crossbone Gundam."

"dress" and between "suit" and "lawsuit", but *not* between "dress" and "lawsuit").

k-NN also gives rise to another problem related to polysemy in DSMs. The problem is that the most frequent senses will populate the top of the nearest neighbor list, while the less frequent senses will not appear until further down the list, and if we set a too restrictive k, we will only see neighbors relating to the most frequent sense. As an example, consider the two different senses of "suit" above. The distributional vector for "suit" can be thought of as a sum $v_{suit} = f_{suit|law}v_{suit|law} + f_{suit|clothes}v_{suit|clothes}$, where $v_{suit|law}$ is an idealized notion of the *true* distributional vector of "suit" in the *law*-sense, and $f_{suit|law}$ is the relative frequency of this sense.[3] From there one can easily argue that a similarity such as $s(v_{suit}, v_{clothes})$ is actually a weighted composite of the similarities $s(v_{suit|law}, v_{clothes})$ and $s(v_{suit|clothes}, v_{clothes})$.[4] If "suit" occurs predominantly in the *law*-sense in our corpus, the k-NN neighborhood of "suit" will be dominated by words pertaining to its *law*-sense, while the less frequent senses might not be present at all. A misguided k may thus obscure any other, less frequent, senses of a term.

3 Word-Sense Induction

Selecting a relevant k for a given term and grouping the neighbors according to which senses they represent is an example of *Word-Sense Induction* (WSI). DSMs are well suited for this task, and there have been a number of different approaches suggested in the literature. One of the earliest approaches is *distributional clustering* (Pereira et al., 1993), which is based on a probabilistic decomposition model that uses maximum likelihood estimation to fit the model to observed data. Another example is *Clustering By Committee* (CBC) (Pantel and Lin, 2002), which first uses average-link clustering to recursively cluster the nearest neighbors of a term into committees, which are then used to define clusters by iteratively adding committees whose similarity to the term exceeds a certain threshold, and that is not too similar to any other added committee. For each added committee, its features are also removed from the distributional representation of the term. This last step ensures that the clusters do not become too similar, and that clusters representing less frequent senses can be discovered.

The idea of iteratively removing features from the distributional vector when a sense cluster as been formed is also present in Dorow and Widdows (2003), who use a graph-based clustering method. Another graph-based approach is the *HyperLex* algorithm (Véronis, 2004), which constructs a graph connecting all pairs of terms that co-occur in the context of an ambiguous term. The resulting graph contains highly connected components, which represent the different senses of the term. Agirre et al. (2006) compare HyperLex to *PageRank* (Brin and Page, 1998) and demonstrates that the two methods perform similarly.

There have also been several attempts to use various types of matrix factorization for WSI. The idea is that the factorization uncovers a set of global senses in the form of the latent factors, and that the sense distribution for a given term can be described as a distribution over these latent factors. Examples of factorization methods that have been used include different versions of *Latent Dirichlet Allocation* ((Brody and Lapata, 2009; Séaghdha and Korhonen, 2011; Yao and Van Durme, 2011; Lau et al., 2012) and *non-negative matrix factorization* (Dinu and Lapata, 2010; Van de Cruys and Apidianaki, 2011).

Tomuro et al. (2007) argue that clustering approaches like distributional clustering or CBC may produce clusters that are themselves polysemous, which may not be a desirable property of a WSI algorithm, and suggests using *feature domain similarity* to solve this problem. The idea is to incorporate similarities between the *features* of items rather than the similarity between the items themselves in a modified version of CBC that enables the algorithm to utilize feature similarities, which inhibit the formation of polysemous clusters.

Koptjevskaja Tamm and Sahlgren (2014) also leverage on the idea of using feature similarity as the basis of sense clustering. The approach, called *syntagmatically labeled partitioning*, relies on a DSM that encodes sequential as well as substitutable relations. The method essentially sorts the k nearest (substitutable) neighbors according to which sequential connections they share. The resulting partitioning of the nearest distributional neighbors does not only constitute a WSI, but it

[3] Weighting schemes muddles this notion quite a bit, but we think the general intuition still holds.

[4] In the case of cosine similarity this follows nicely from the distributive property of dot products: $v = av_1 + bv_2$, $s(v, w) = \frac{v \cdot w}{\|v\|\|w\|} = \frac{a(v_1 \cdot w) + b(v_2 \cdot w)}{\|v\|\|w\|}$

also provides *labels* for the induced senses in the form of the sequential connections the neighbors share.

4 Neighborhood Graphs

Many of the previous WSI approaches operate at a global level, utilizing global structural properties of the semantic spaces, e.g. by matrix factorization techniques. We believe this is as ill-advised as setting a global k or radius for the nearest neighbor search, since it is the *local* structures that are important when analyzing nearest neighbors. Other WSI approaches use various forms of clustering techniques. However, previous studies of the intrinsic dimensionality of distributional semantic spaces using fractal dimensions indicate that neighborhoods in semantic space have a *filamentary* rather than clustered structure (Karlgren et al., 2008).

We therefore propose the use of *topological* models that take the *local* structure of neighborhoods in semantic space into account. The approach discovers different word senses from the local structure of neighborhoods, given nothing but similarities between points. As such it is easy to test on widely different vector models, as long as there exists a well behaved similarity function. The proposed approach not only answers the question which other terms are similar to a given term, but also *how* are they similar.

Relative neighborhoods, first proposed in (Toussaint, 1980), are examples of *empty region graphs* (Cardinal et al., 2009), where points are neighbors if some region between them is empty. For *Relative Neighborhood Graphs* (RNG) this region between two points a and c is defined as the intersection of the two spheres with centers in a and c with radius $d(a, c)$. In other words, a point b lies between points a and c if it is closer to both a and c than a and c are to each other. If no such point b exists, a and c are neighbors. Illustrations of this can be seen in Figure 1.

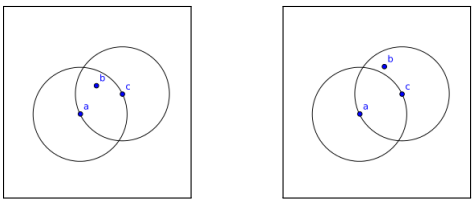

Figure 1: Example of when point b is between point a and c (left), and when it is not (right).

Such neighborhoods have been argued to better preserve local topology (Bremer et al., 2014), and be more robust to deformations of the data than k-NN neighborhoods (Correa and Lindstrom, 2012) as they in some sense contain information about direction whereas k-NN neighborhoods only contain information about distance. Going back to the "suit" example, we can see that if "suit" in the *law*-sense is more similar to the composite "suit" than to its *clothes*-sense, and vice versa, then the composite v_{suit} lies between $v_{suit|law}$ and $v_{suit|clothes}$. This in turn means that out of those two points, both are relative neighbors to "suit", and neither of them lies between the other and "suit".

Formally, the set of points between two points $a, c \in V$ can be characterized and computed in the following way:

$$\text{btw}(V, a, c) = \{b | b \in V, b \text{ is between } a \text{ and } c\}$$

$$\text{rng-nbh}(V, a) = \{c | c \in V, \text{btw}(V, a, c) = \emptyset\}$$

$$E_{\text{rng}}(V) = \{(a, b) | a \in V, b \in \text{rng-nbh}(V, a)\}$$

where E_{rng} is the undirected edge set of the RNG. The function $\text{btw}(V, a, c)$ can be straightforwardly translated to an algorithm taking $\mathcal{O}(|V|)$ time, making the $\text{rng-nbh}(V, a)$ function take $\mathcal{O}(|V|^2)$ time, which in turn makes the computation of the complete graph take $\mathcal{O}(|V|^3)$ time.[5] Clearly unfeasible, but we have not found any alternatives that perform better in the high-dimensional case.[6]

Correa and Lindstrom (2012) note that the intersection of the RNG and the k-NN graph is a more feasible alternative:

$$\text{k-rng-nbh}(V, a) = \text{rng-nbh}(V', a)$$

$$\text{where } V' = k \text{ nearest neighbors of } a.$$

Given a precompiled k-NN lookup, the above takes $\mathcal{O}(k^2)$ time, so using a heap-based $\mathcal{O}(|V| \lg k)$ k-NN algorithm results in an algorithm taking $\mathcal{O}(k^2 + |V| \lg k)$ time.

The same idea can be used to build a tree structure rooted in a reference word a in the following way:

$$\text{rnbh-tree}(V, a) = \{(c, \arg\min_{b \in B_c} d(b, c)) | c \in V\}$$

$$\text{where } B_c = \{a\} \cup \text{btw}(V, a, c)$$

[5] Assuming a constant time distance function.

[6] It should be noted that there are more efficient algorithms for lower-dimensional situations.

which can easily be restricted to the k-nearest neighbors of a in much the same way as above, with the same monotonic behavior.

Computing this for a point a produces a tree where the direct children of a are its relative neighbors, and the parent of a point c further down the tree is the point between a and c that is closest to c. This structure, while similar to a *minimum spanning tree*, differs in some crucial regards: the rnbh-tree(V, a) is rooted in a word a. The difference between rnbh-tree(V, a) and rnbh-tree(V, b) is often quite significant. Furthermore, the restricted k-rnbh-tree is monotonic in k. That property does not hold for a minimum spanning tree of a local neighborhood.

5 Examples of RNGs

To get an intuition of what these neighborhoods look like we present a few examples. The words have been chosen either because they are common examples in similar work — e.g. "heart" and "suit" from Pantel and Lin (2002) — or because they represent different parts-of-speech ("above" is a preposition, "bad" is an adjective, and "service" is a noun) and disparate kinds of ambiguity ("orange" can be both a fruit and a color).

Figure 2 (next page) illustrates what an RNG looks like for the term "heart" and its 100 nearest neighbors in the PMI model. Note that the root "heart" (at the mid-left in the graph) only has two relative neighbors: "cardiac" and "soul," arguably representing a *body*-sense and a *soul*-sense of the term. One advantage of using this type of structure for the neighborhood is that it enables us to examine various depths of the tree. Depth one includes only the direct neighbors ("cardiac" and "soul"), while depth two includes all neighbors two steps away in the graph: "disease," "coronary," "pulmonary," "cardiovascular," "ventricular," and "failure," which are all children to "cardiac." This tree structure can be used to identify neighbors that are themselves polysemous (c.f. the critique mentioned in Section 3 of clustering-based approaches to word-sense induction that they may produce polysemous clusters). One example is the neighbor "disease" at depth two, which has six children that refer to different aspects of disease.

We argue that the RNG can be quite useful for WSI, since the branching structure indicates different usages, and the depth factor enables us to calibrate the granularity of the induced word senses. If we only consider direct neighbors (i.e. depth one), and set $k = V$ (i.e. we do an exhaustive nearest neighbor search), we will extract all terms that have a direct connection to the reference term. We refer to this neighborhood as the *semantic horizon*. At the most coarse level of analysis, this is the neighborhood that represents the main induced senses of a term. Tables 2 and 3 provide examples of 1,000-RNG neighborhoods of depth one.

Table 2: RNG for $k = 1,000$ of the words "suit," "orange," and "heart" in three different semantic models. The numbers in parenthesis indicate the k-NN ranks of the neighbors.

PMI	**GloVe**	**Skipgram**
suit		
suits (1) dress (2) lawsuit (10) dinosaur (53) costly (60) option (76) counterparts (99) predator (107) trump (109) ⋮	suits (1) lawsuit (2) mobile (33) gundam (34) trump (55) zoot (133) rebid (423) serenaders (458) hev (987)	suits (1) lawsuit (2)
orange		
yellow (1) lemon (16)	yellow (1) ktype (12) lemon (14) citrus (17) jersey (21) cherry (24) county (26) peel (42) jumpsuits (57) ⋮	redorange (1)
heart		
cardiac (1) soul (22) hearts (183) ashtray(641) rags(771)	my (1) blood (2) throbs (3) suffering (4) brain (6) cardiac (8) hearts (11) throb (17) lungs (22) ⋮	congestive (1) hearts (2)

These examples demonstrate some interesting similarities and differences between the three models. First of all, there are some direct neighbors that are present in all three models: "suit" has "suits" and "lawsuit" as direct neighbors in all three models, "heart" has "hearts," "service"

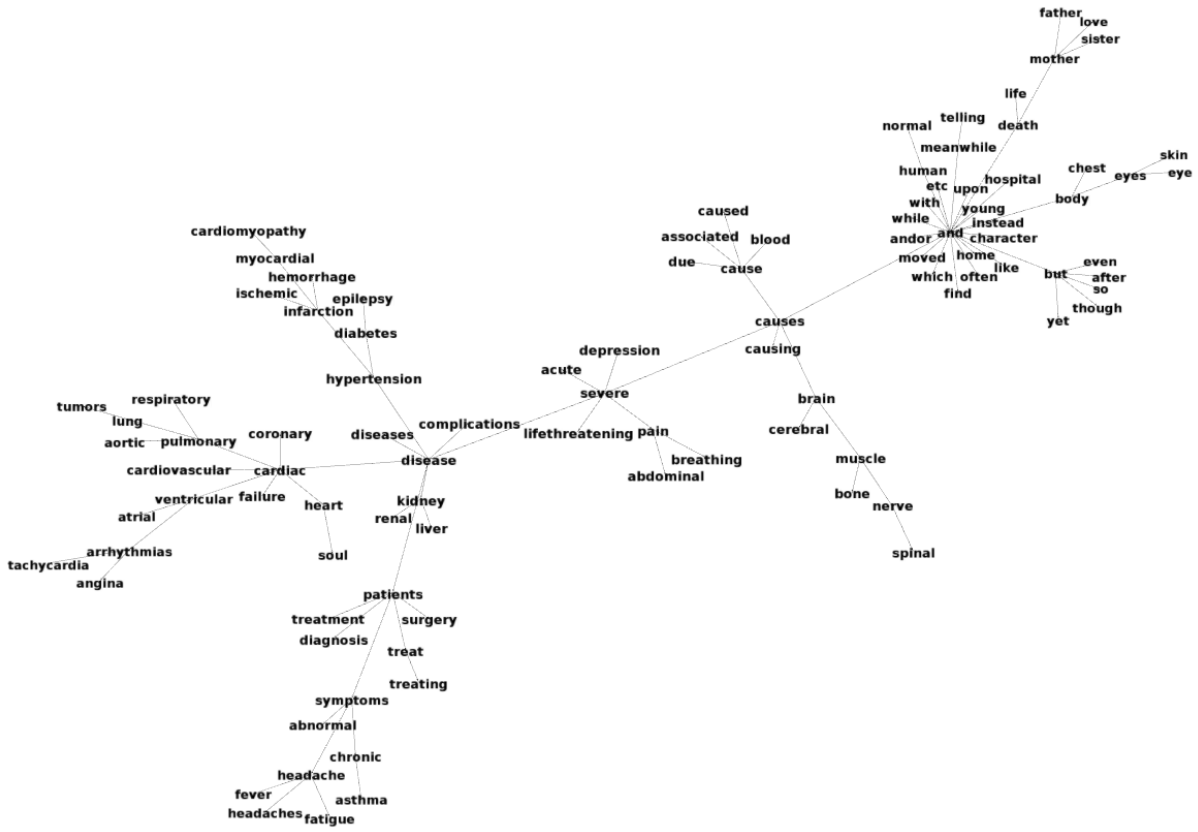

Figure 2: RNG for “heart” in the PMI model, restricted to the 100 nearest neighbors.

has “services,” and “above” has “below”. Plural forms are of course reasonable neighbors of their singular counterparts in a semantic model, but their usefulness for WSI can perhaps be questioned. Taking “suits” to indicate the *clothes*-sense of “suit,” all three models produce both a *clothes*-sense and a *law*-sense. For “orange,” the Skipgram model only represents the *color*-sense, while the PMI and GloVe models also feature a *fruit*-sense. For “heart,” all three models have a *disease*-sense (represented by the neighbors “cardiac” in the PMI and GloVe models, and the neighbor “congestive” in the Skipgram model), and an *organ*-sense (represented by the plural form “hearts”). “Service” is a comparably vague term that has a number of different senses in the PMI and GloVe models, but only one in the Skipgram model. “Bad” produces both a *negativity*-sense and a *German spa town*-sense in all three models, but only the GloVe and Skipgram models have a separate *antonym*-sense (“good” is not a direct neighbor in the PMI model). “Above” has both the antonym and direct neighbors relating to measurements in all three models.

It is interesting to note that GloVe produces a significant amount of sequential relations; “mobile suit gundam”, “cheap suit serenaders”, “orange peel”, and “orange jumpsuit” are just some of many examples of sequential relations found in the relative neighborhood of terms in the GloVe model.

The PMI and GloVe models produce the structurally most similar RNGs in these examples, with on average a handful of direct neighbors, of which some can be very distant. The Skipgram model on the other hand produces very few direct neighbors. This led us to look further into the structural properties of neighborhoods in the Skipgram model. An interesting observation — and possible complication — is that the neighborhoods in the Skipgram model are highly asymmetric: the first neighbor of “information” is “informations”, whereas “information” is the 1,829th neighbor of “informations.” While such asymmetry occurs in all models, it seems much more prevalent in the Skipgram model. Figure 3 confirms this suspicion: each point corresponds to a random word pair (a, b) with x corresponding to where b is in the ordered list of a’s neighbor, and y to where a is in the ordered list of b’s neighbors. The figure

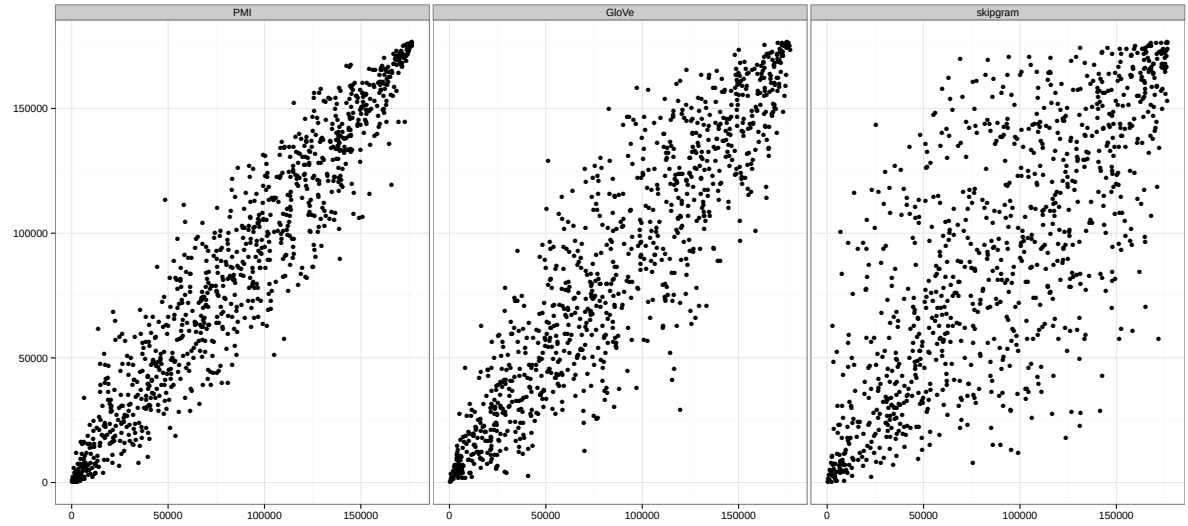

Figure 3: Neighborhood reciprocity in the different models; PMI to the left, GloVe in the middle, and Skipgram to the right.

Table 3: k-RNG for $k = 1,000$ of the words "service," "bad," and "above" in three different semantic models. The numbers in parenthesis indicate the k-NN ranks of the neighbors.

PMI	**GloVe**	**Skipgram**
service		
services (1) network (2) operates (8) launched (18) served (22) intercity(34)	services (1) operated (3) serving (6) military (17) duty (20) passenger (21) dialaride (644) aftersales (759) limitedstop (802)	services (1)
bad		
terrible (1) that (2) luck (39) unfortunate (70) stalling (276) donnersbergkreis (860) rancid (980)	good (1) kissingen (2) ugly (45) nasty (48) dirty (106) omen (328) conkers (360) karma (952)	nauheim (1) good (2) dreadful (5)
above		
below (1) around (2) feet (5) measuring (29) beneath (36) columns (62) atop (102)	below (1) level (2) height (3) just (4) stands (10) lower (11) beneath (12) rise (21) sea (30) ⋮	below (1) 500ft (2)

shows that the local densities vary much more in the Skipgram model than in the others. This is not in itself undesirable, but wild differences in neighborhood reciprocity complicates the choice of k in the k-RNG algorithm, as observed by the particularly sparse neighborhoods of the Skipgram model above.

6 WSI Evaluation

The standard way to evaluate WSI algorithms is to use one the SemEval WSI test collections (Agirre and Soroa, 2007; Manandhar et al., 2010; Navigli and Vannella, 2013; Jurgens and Klapaftis, 2013), which are all designed similarly: systems are expected to first perform WSI and then to assign texts to the induced senses (i.e. in effect doing a word-sense *disambiguation* step). We consider this type of evaluation to be a less useful for our purposes, since the required disambiguation step is a highly non-trivial task in itself. The RNG method proposed in this paper is a pure WSI algorithm, and as such does not offer a solution to the disambiguation problem. We therefore opted to focus solely on the hypothesis that relative neighborhoods cover senses that k-NNs do not. In essence, we investigate whether k-RNG *retrieval* does a better job at covering different senses than k-NN *retrieval*. This was done using *pseudowords*.

Pseudowords are artificially ambiguous words, created by regarding different words as identical. We can, for example, say that the pseu-

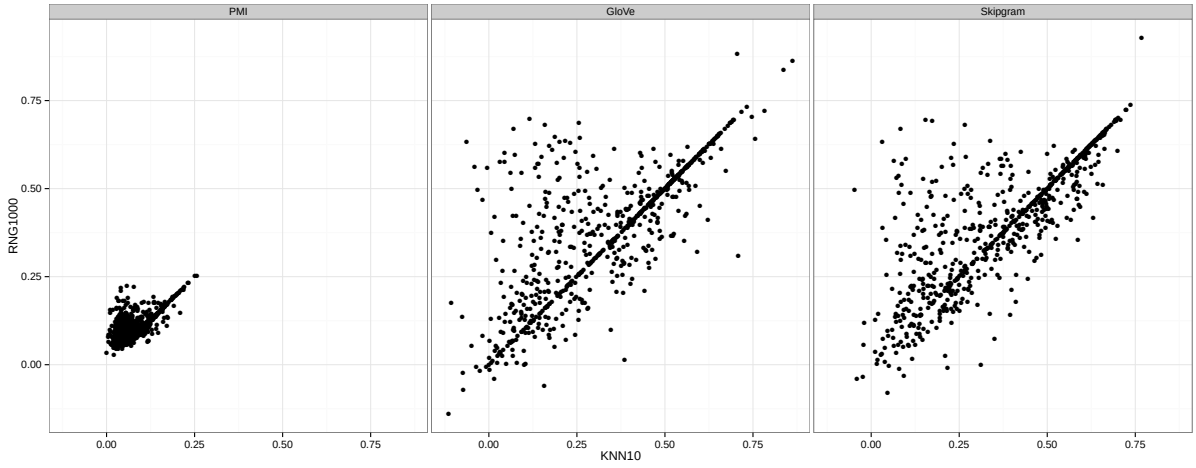

Figure 4: Comparison of minmax pseudosense score for k-RNGs and k-NNs for $k = 1,000$ and $k = 10$ respectively; PMI to the left, GloVe in the middle, and Skipgram to the right.

doword <*deadeye*> is a composite of the two words *marksman* and *loudspeaker*. A corpus with the artificially ambiguous word <*deadeye*> in it can then be created by replacing all occurrences of the words *marksman* and *loudspeaker* with <*deadeye*>.

Using the pseudowords provided by Pilehvar and Navigli (2013) a corpus with 689 non-overlapping pseudowords was created, based on the BNC corpus.[7] Two models were then trained, one on the altered corpus, and one on the unaltered one. To check whether the neighborhood of a pseudoword contains information about its underlying senses we compared each underlying sense to the words in the neighborhood, taking the minimum of all senses' maximum similarity as a score, as demonstrated in Table 4. The similarities were calculated using the model trained on the unaltered corpus, as the one based on the altered corpus will not contain the underlying senses of pseudowords.

Working through the example in Table 4, the neighborhood of the pseudoword <*deadeye* > consists of the three words *shooter*, *stereo*, and *sport*. The pseudoword in itself is made up of the two underlying senses *marksman* and *loudspeaker*. The similarities between the words in the neighborhood of the model trained on the unaltered data and the words of the underlying senses are as presented in Table 4. The closest word to *marksman* is *shooter*, with a similarity score of 0.7. The closest word to *loudspeaker* is *stereo*, with a score of 0.3. So the scoring would, in total, be 0.3. It should be noted that the upper bound for this score is oftentimes significantly lower than 1: The neighborhood could not possibly contain the words *marksman* or *loudspeaker*, as those words are not present in the corpus. This means that the scores are bounded by the similarity of the least similar closest neighbor to the underlying senses.

Table 4: Example scoring of a neighborhood of the word <*deadeye*>.

<*deadeye*>	*shooter*	*stereo*	*sport*	**max**
marksman	0.7	0.04	0.4	0.7
loudspeaker	0.01	0.3	0.05	0.3
min: 0.3				

This score was chosen because of its simplicity and intuitive interpretation: a low score implies that at least one word sense was not represented in the neighborhood whereas a high score means that all senses are represented in the neighborhood. One can then plot these scores for both relative neighborhoods and k-NN neighborhoods for each pseudoword as is done in Figure 4. Each

[7] www.natcorp.ox.ac.uk

point (x, y) represents a pseudoword, with x and y being the score of the k-NN neighborhood and the k-RNG neighborhood respectively.

Figure 5 shows an aggregate of Figure 4, plotting the distribution of $y - x$, i.e. the difference between the scores achieved by the k-RNG and the k-NN. As seen in Figure 4, a lot of points lie on the line $y = x$, meaning both methods achieved the same score. However, when this is not the case, there is a clear bias for the k-RNG to outperform the k-NN, as demonstrated in Figure 5. Here, using the BNC instead of Wikipedia as training data, the GloVe and Skipgram models yielded sparse relative neighborhoods — both with an average of about 8 neighbors — but the PMI model produced quite dense neighborhoods averaging 63 neighbors. Since the scoring function does not penalize neighborhood size there is good reason to be skeptical of its viability, and specifically the performance of the PMI-model based on these figures.

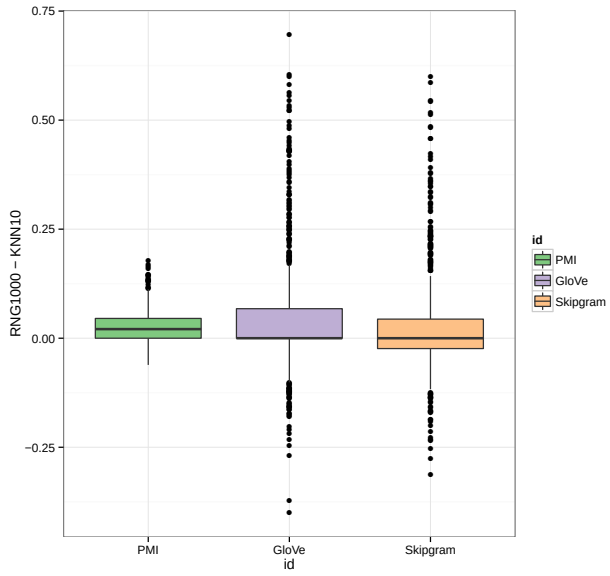

Figure 5: Distribution of difference between scores for k-RNGs and k-NNs. Positive scores means that the k-RNG scored higher than the k-NN

7 Conclusions

This paper has discussed the question how to query semantic models, which is a question that has been long neglected in research on computational semantics. Nearest neighbor search (or k-NN) is often treated as the only available option, which leads to misunderstandings regarding how semantic models represent and handle vagueness and polysemy. We have argued that the structure — or topology — of the local neighborhoods in semantic models carry useful semantic information regarding the different usages — or senses — of a term, and that such topological properties therefore can be used to analyze polysemy and do WSI.

We have introduced relative neighborhood graphs (RNG) as an alternative to standard k-NN, and we have exemplified k-RNG in three different well-known semantic models. The examples demonstrate that k-RNG manages to retrieve disparate and relevant neighbors in all three models, yet the kind of neighbors returned and the nature of the neighborhoods differ. Quantitatively, The k-RNG method consistently outperformed k-NN on underlying sense retrieval.

We have also illustrated how k-RNG can be used as a tool to gain insight into the topological properties of different models. The GloVe model, for example, makes no difference between sequential and substitutable relations, leading to neighborhoods that contain n-grams instead of senses. This can clearly be seen in for example Table 2. Skipgram uses more sophisticated tokenization, which alleviates this issue.

Another interesting result of the paper is that the RNG uncovers otherwise unseen differences between the models, which manifest not as scoring differences but as properties of the word representations themselves. One example is the differences in neighborhood reciprocity observed between the different models.

References

Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of SemEval*, pages 7–12.

Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Two graph-based algorithms for state-of-the-art WSD. In *Proceedings of EMNLP*, pages 585–593.

Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923.

Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

Peer-Timo Bremer, Ingrid Hotz, Valerio Pascucci, and Ronald Peikert. 2014. *Topological Methods in Data Analysis and Visualization III*. Springer.

Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of WWW*, pages 107–117.

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of EACL*, pages 103–111.

Jean Cardinal, Sébastien Collette, and Stefan Langerman. 2009. Empty region graphs. *Computational geometry*, 42(3):183–195.

Carlos D Correa and Peter Lindstrom. 2012. Locally-scaled spectral clustering using empty region graphs. In *Proceedings of KDD*, pages 1330–1338.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of EMNLP*, pages 1162–1172.

Beate Dorow and Dominic Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of EACL*, pages 79–82.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of ACL*, pages 92–97.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of STOC*, pages 604–613.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Proceedings of SemEval*, pages 290–299.

Jussi Karlgren, Anders Holst, and Magnus Sahlgren. 2008. Filaments of meaning in word space. In *Proceedings of ECIR*, pages 531–538.

Maria Koptjevskaja Tamm and Magnus Sahlgren. 2014. Temperature in word space. In Benedikt Szmrecsanyi and Bernhard Wälchli, editors, *Aggregating dialectology, typology, and register analysis*, pages 231–267. De Gruyter.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of EACL*, pages 591–601.

Suresh Manandhar, Ioannis P. Klapaftis, Dmitriy Dligach, and Sameer S. Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of SemEval*, pages 63–68.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Roberto Navigli and Daniele Vannella. 2013. Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application. In *Proceedings of SemEval*, pages 193–201.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of KDD*, pages 613–619.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of ACL*, pages 183–190.

Mohammad Taher Pilehvar and Roberto Navigli. 2013. Paving the way to a large-scale pseudosense-annotated dataset. In *Proceedings of NAACL-HLT*, pages 1100–1109.

Diarmuid Ó Séaghdha and Anna Korhonen. 2011. Probabilistic models of similarity in syntactic context. In *Proceedings of EMNLP*, pages 1047–1057.

Noriko Tomuro, Steven L. Lytinen, Kyoko Kanzaki, and Hitoshi Isahara. 2007. Clustering using feature domain similarity to discover word senses for adjectives. In *Proceedings of ICSC*, pages 370–377.

Godfried T. Toussaint. 1980. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261 – 268.

Tim Van de Cruys and Marianna Apidianaki. 2011. Latent semantic word sense induction and disambiguation. In *Proceedings of HLT*, pages 1476–1485.

Jean Véronis. 2004. HyperLex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

Xuchen Yao and Benjamin Van Durme. 2011. Nonparametric bayesian word sense induction. In *Proceedings of TextGraphs*, pages 10–14.