Robust Handling of Polysemy via Sparse Representations

Abhijit A. Mahabal
Google
amahabal@google.com

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

Sid Mittal
Google
sidmittal@google.com

Abstract

Words are polysemous and multi-faceted, with many shades of meanings. We suggest that sparse distributed representations are more suitable than other, commonly used, (dense) representations to express these multiple facets, and present *Category Builder*, a working system that, as we show, makes use of sparse representations to support multi-faceted lexical representations. We argue that the set expansion task is well suited to study these meaning distinctions since a word may belong to multiple sets with a different reason for membership in each. We therefore exhibit the performance of *Category Builder* on this task, while showing that our representation captures at the same time *analogy* problems such as "the Ganga of Egypt" or "the Voldemort of Tolkien". *Category Builder* is shown to be a more expressive lexical representation and to outperform dense representations such as Word2Vec in some analogy classes despite being shown only two of the three input terms.

1 Introduction

Word embeddings have received much attention lately because of their ability to represent similar words as nearby points in a vector space, thus supporting better generalization when comparisons of lexical items are needed, and boosting the robustness achieved by some deep-learning systems. However, a given surface form often has multiple meanings, complicating this simple picture. Arora et al. (2016) showed that the vector corresponding to a polysemous term often is not close to any of that of its individual senses, thereby breaking the similar-items-map-to-nearby-points promise. The polysemy wrinkle is not merely an irritation but, in the words of Pustejovsky and Boguraev (1997), "one of the most intractable problems for language processing studies".

Our notion of Polysemy here is quite broad, since words can be similar to one another along a variety of dimensions. The following three pairs each has two similar items: (a) {*ring, necklace*}, (b) {*ring, gang*}, and (c) {*ring, beep*}. Note that ring is similar to all words that appear as second words in these pairs, but for different reasons, *defined by the second token* in the pairs. While this example used different senses of *ring*, it is easy to find examples where a single sense has multiple *facets*: *Clint Eastwood*, who is both an actor and a director, shares different aspects with directors than with actors, and *Google*, both a website and a major corporation, is similar to *Wikipedia* and *General Electric* along different dimensions.

Similarity has typically been studied pairwise: that is, by asking how similar item *A* is to item *B*. A simple modification sharply brings to fore the issues of facets and polysemy. This modification is best viewed through the task of *set expansion* (Wang and Cohen, 2007; Davidov et al., 2007; Jindal and Roth, 2011), which highlights the similarity of an item (a candidate in the expansion) to a set of seeds in the list. Given a few seeds (say, {*Ford, Nixon*}), what else belongs in the set? Note how this expansion is quite different from the expansion of {*Ford, Chevy*}, and the difference is one of *Similar How*, since whether a word (say, *BMW* or *FDR*) belongs in the expansion depends not just on how much commonality it shares with *Ford* but on *what* commonality it shares. Consequently, this task allows the same surface form to belong to multiple sets, by virtue of being similar to items in distinct sets *for different reasons*. The facets along which items are similar is implicitly defined by the members in the set.

In this paper, we propose a context sensitive version of similarity based on highlighting shared facets. We do this by developing a *sparse representation* of words that simultaneously captures all

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 265–275
New Orleans, June 5-6, 2018.

facets of a given surface form. This allows us to define a notion of contextual similarity, in which *Ford* is similar to *Chevy* (e.g., when *Audi* or *BMW* is in the context) but similar to *Obama* when *Bush* or *Nixon* is in the context (i.e., in the seed list). In fact, it can even support multi-granular similarity since while {*Chevy, Chrysler, Ford*} represent the facet of AMERICAN CARS, {*Chevy, Audi, Ford*} define that of CARS. Our contextual similarity is better able to mold itself to this variety since it moves away from the one-size-fits-all nature of cosine similarity.

We exhibit the strength of the representation and the contextual similarity metric we develop by comparing its performance on both set expansion and analogy problems with dense representations.

2 Senses and Facets

The present work does not attempt to resolve the Word Sense Disambiguation (WSD) problem. Rather, our goal is to advance a lexical representation and a corresponding context sensitive similarity metric that, together, get around explicitly solving WSD.

Polysemy is intimately tied to the well-explored field of WSD so it is natural to expect techniques from WSD to be relevant. If WSD could neatly separate senses, the set expansion problem could be approached thus. *Ford* would split into, say, two senses: *Ford-1* for the car, and *Ford-2* for the president, and expanding {*Ford, Nixon*} could be translated to expanding {*Ford-2, Nixon*}. Such a representational approach is taken by many authors when they embed the different senses of words as distinct points in an embedding space (Reisinger and Mooney, 2010; Huang et al., 2012; Neelakantan et al., 2014; Li and Jurafsky, 2015).

Such approaches run into what we term *the Fixed Inventory Problem.* Either senses are obtained from a hand curated resource such as a dictionary, or are induced from the corpus directly by mapping contexts clusters to different senses. In either case, however, by the time the final representation (e.g., the embedding) is obtained, the number of different senses of each term has become fixed: all decisions have been made relating to how finely or coarsely to split senses.

How to split senses is a hard problem: dictionaries such as NOAD list coarse senses and split these further into fine senses, and it is unclear what granularity to use: should each fine sense correspond to a point in the vector space, or should, instead, each coarse sense map to a point? Many authors (Hofstadter and Sander, 2013, for example) discuss how the various dictionary senses of a term are not independent. Further, if context clusters map to senses, the word *whale*, which is seen both in mammal-like contexts (e.g., "whales have hair") and water-animal contexts ("whales swim"), could get split into separate points. Thus, the different *senses* that terms are split into may instead be distinct *facets*. This is not an idle theoretical worry: such facet-based splitting is evident in Neelakantan et al. (2014, Table 3). Similarly, in the vectors they released, *november* splits into ten senses, likely based on facets. Once split, for subsequent processing, the points are independent.

In contrast to such explicit, prior, splitting, in the Category Builder approach developed here, relevant contexts are chosen given the task at hand, and if multiple facets are relevant (as happens, for example, in {*whale, dolphin, seal*}, whose expansion should rank aquatic mammals highest), all these facets influence the expansion; if only one facet is of relevance (as happens in {*whale, shark, seahorse*}), the irrelevant facets get ignored.

3 Related Work

In this section, we situate our approach within the relevant research landscape. Both *Set Expansion* and *Analogies* have a long history, and both depend on *Similarity*, with an even longer history.

3.1 Set Expansion

Set Expansion is the well studied problem of expanding a given set of terms by finding other semantically related terms. Solutions fall into two large families, differing on whether the expansion is based on a preprocessed, limited corpus (Shen et al., 2017, for example) or whether a much larger corpus (such as the entire web) is accessed on demand by making use of a search engine such as Google (Wang and Cohen, 2007, for example).

Each family has its advantages and disadvantages. "Open web" techniques that piggyback on Google can have coverage deep into the tail. These typically rely on some form of Wrapper Induction, and tend to work better for sets whose instances show up in lists or other repeated structure on the web, and thus perform much better on sets of nouns than on sets of verbs or adjectives. By contrast, "packaged" techniques that work off a

preprocessed corpus are faster (no Google lookup needed) and can work well for any part of speech, but are of course limited to the corpus used. These typically use some form of distributional similarity, which can compute similarity between items that have never been seen together in the same document; approaches based on shared memberships in lists would need a sequence of overlapping lists to achieve this. Our work is in the "packaged" family, and we use sparse representations used for distributional similarity.

Gyllensten and Sahlgren (2018) compares two subfamilies within the packaged family: *centrality*-based methods use a prototype of the seeds (say, the centroid) as a proxy for the entire seed set and *classification*-based methods (a strict superset), which produce a classifier by using the seeds. Our approach is classification-based.

It is our goal to be able to expand nuanced categories. For example, we want our solution to expand the set {*pluto, mickey*}—both Disney characters—to other Disney characters. That is, the context *mickey* should determine what is considered 'similar' to pluto, rather than being biased by the more dominant sense of *pluto*, to determine that *neptune* is similar to it. Earlier approaches such as Rong et al. (2016) approach this problem differently: they expand to both planets and Disney characters, and then attempt to cluster the expansion into meaningful clusters.

3.2 Analogies

Solving analogy problems usually refers to proportional analogies, such as *hand*:*glove*::*foot*:*?*. Mikolov et al. (2013) showed how word embeddings such as Word2Vec capture linguistic regularities and thereby solve this. Turney (2012) used a pair of similarity functions (one for *function* and one for *domain*) to address the same problem.

There is a sense, however, that the problem is *overdetermined*: in many such problems, people can solve it even if the first term is not shown. That is, people easily answer "What is the *glove* for the *foot*?". People also answer questions such as "What is the Ganga of Egypt?" without first having to figure out the unprovided term *India* (or is the missing term *Asia*? It doesn't matter.) Hofstadter and Sander (2013) discuss how our ability to do these analogies is central to cognition.

The current work aims to tackle these *non-proportional* analogies and in fact performs better than Word2Vec on some analogy classes used by Mikolov et al. (2013), despite being shown one fewer term.

The approach is rather close to that used by Turney (2012) for a different problem: *word compounds*. Understanding what a *dog house* is can be phrased as "What is the house of a dog?", with *kennel* being the correct answer. This is solved using the pair of similarity functions mentioned above. The evaluations provided in that paper are for *ranking*: which of five provided terms is a match. Here, we apply it to non-proportional analogies and evaluate for retrieval, where we are ranking over all words, a significantly more challenging problem.

To our knowledge, no one has presented a computational model for analogies where only two terms are provided. We note, however, that Linzen (2016) briefly discusses this problem.

3.3 Similarity

Both Set Expansion and Analogies depend on a notion of similarity. Set Expansion can be seen as finding items most similar to a category, and Analogies can be seen as directly dependent on similarities (e.g., in the work of Turney (2012)).

Most current approaches, such as word embeddings, produce a context independent similarity. In such an approach, the similarity between, say, *king* and *twin* is some fixed value (such as their cosine similarity). However, depending on whether we are talking about bed sizes, these two items are either closely related or completely unrelated, and thus context dependent.

Psychologists and Philosophers of Language have long pointed out that similarity is subtle. It is sensitive to context and subject to priming effects. Even the very act of categorization can change the perceived similarity between items (Goldstone et al., 2001). Medin et al. (1993, p. 275) tell a story, from the experimental psychology trenches, that supports representation morphing when they conclude that "the effective representations of constituents are determined in the context of the comparison, not prior to it".

Here we present a malleable notion of similarity that can adapt to the wide range of human categories, some of which are based on narrow, superficial similarities (e.g., BLUE THINGS) while others share family resemblances (à la Wittgenstein). Even in a small domain such as movies, in differ-

ent contexts, similarity may be driven by who the director is, or the cast, or the awards won. Furthermore, to the extent that the contexts we use are human readable, we also have a mechanism for explaining what makes the terms similar.

There is a lot of work on the context-dependence of human categories and similarities in Philosophy, in Cognitive Anthropology and in Experimental Psychology (Lakoff, 1987; Ellis, 1993; Agar, 1994; Goldstone et al., 2001; Hofstadter and Sander, 2013, for example, survey this space from various theoretical standpoints), but there are not, to our knowledge, unsupervised computational models of these phenomena.

4 Representations and Algorithms

This section describes the representation and corresponding algorithms that perform set expansion in Category Builder (CB).

4.1 Sparse Representations for Expansion

We use the traditional word representation that distributional similarity uses (Turney and Pantel, 2010), and that is commonly used in fields such as context sensitive spelling correction and grammatical correction (Golding and Roth, 1999; Rozovskaya and Roth, 2014); namely,words are associated with some ngrams that capture the contexts in which they occur – all contexts are represented in a sparse vector corresponding to a word. Following Levy and Goldberg (2014a), we call this representation *explicit*.

Generating Representations. We start with web pages and extract words and phrases from these, as well as the contexts they appear in. An aggregation step then calculates the strengths of word to context and context to word associations.

Vocabulary. The vocabulary is made up of words (nouns, verbs, adjectives, and adverbs) and some multi-word phrases. To go beyond words, we use a named entity recognizer to find multi-word phrases such as *New York*. We also use one heuristic rule to add certain phrasal verbs (e.g., *take shower*), when a verb is directly followed by its direct object. We lowercase all phrases, and drop those phrases seen infrequently. The set of all words is called the vocabulary, $\mathcal{V}$.

Contexts. Many kinds of contexts have been used in literature. Levy (2018) provides a comprehensive overview. We use contexts derived from syntactic parse trees using about a dozen heuristic rules. For instance, one rule deals with nouns modified by an adjective, say, *red* followed by *car*. Here, one of the contexts of *car* is MODIFIEDBY#RED, and one of the contexts of *red* is MODIFIES#CAR. Two more examples of contexts: OBJECTOF#EAT and SUBJECTOF#WRITE. The set of all contexts is denoted $\mathcal{C}$.

The Two Vocabulary⇔Context matrices. For vocabulary $\mathcal{V}$ and contexts C, we produce *two* matrices, $M^{\mathcal{V}\rightarrow\mathcal{C}}$ and $M^{\mathcal{C}\rightarrow\mathcal{V}}$. Many measures of association between a word and a context have been explored in the literature, usually based on some variant of *pointwise mutual information*.

PPMI (*Positive* PMI) is the typically used measure. If $P(w)$, $P(c)$ and $P(w,c)$ respectively represent the probabilities that a word is seen, a context is seen and the word is seen in that context, then

$$\text{PMI}(w,c) = \log \frac{P(w,c)}{P(w)P(c)} \tag{1}$$

$$\text{PPMI}(w,c) = max(0, \text{PMI}(w,c)) \tag{2}$$

PPMI is widely used, but comments are in order regarding the ad-hocness of the "0" in Equation 2. There is seemingly a good reason to choose 0 as a threshold: if a word is seen in a context more than by chance, the PMI is positive, and a 0 threshold seems sensible. However, in the presence of polysemy, especially lopsided polysemy such as *Cancer* (disease and star sign), a "0" threshold is arbitrary: even if every single occurrence of the star sign sense of *cancer* was seen in some context c (thereby crying out for a high PMI), because of the rarity of that sense, the overall PMI between c and (non-disambiguated) *Cancer* may well be negative. Relatedly, Shifted PPMI (Levy and Goldberg, 2014b) uses a non-0 cutoff.

Another well known problem with PPMI is its large value when the word or the context is rare, and even a single occurrence of a word-context pair can bloat the PMI (see Role and Nadif, 2011, for fixes that have been proposed). We introduce a new variant we call *Asymmetric PMI*, which takes frequency into account by adding a second log term, and is asymmetric because in general $P(w|c) \neq P(c|w)$:

$$\begin{aligned}\text{APMI}(w,c) &= \text{PMI}(w,c) + \log \frac{P(w,c)}{P(w)} \\ &= \log \frac{P(w,c)^2}{P(w)^2 P(c)}\end{aligned} \tag{3}$$

This is asymmetric because $\mathrm{APMI}(c, w)$ has $P(c)$ in the denominator of the extra log term.

What benefit does this modification to PMI provide? Consider a word and two associated contexts, c_1 and c_2, where the second context is significantly rarer. Further, imagine that the PMI of the word with either feature is the same. The word would have been seen in the rarer context only a few times, and this is more likely to have been a statistical fluke. In this case, the APMI with the more frequent term is higher: we reward the fact that the PMI is high despite its prevalence; this is less likely to be an artifact of chance.

Note that the rearranged expression seen in the second line of Equation 3 is reminiscent of $\mathrm{PPMI}^{0.75}$ from Levy et al. (2015).

The second log term in APMI is always negative, and we thus shift all values by a constant k (chosen based on practical considerations of data size: the smaller the k, the larger the size of the sparse matrices; based on experimenting with various values of k, it appears that expansion quality is not very sensitive to k). Clipping this shifted value at 0 produces Asymmetrical PPMI (APPMI):

$$\mathrm{APPMI}(w, c) = \max(0, \mathrm{APMI}(w, c) + k) \quad (4)$$

The two matrices thus produced are shown in Equation 5. If we use PPMI instead of APPMI, these are transposes of each other.

$$\begin{aligned} M^{\mathcal{V}\to\mathcal{C}}_{w,c} &= \mathrm{APPMI}(w, c) \\ M^{\mathcal{C}\to\mathcal{V}}_{c,w} &= \mathrm{APPMI}(c, w) \end{aligned} \quad (5)$$

4.2 Focused Similarity and Set Expansion

We now come to the central idea of this paper: the notion of focused similarity. Typically, similarity is based on the dot product or cosine similarity of the context vectors. The pairwise similarity among all terms can be expressed as a matrix multiplication as shown in Equation 6. Note that if we had used PPMI in Equation 5, the matrices would be each other's transposes and each entry in $\mathrm{SimMatrix}$ in Equation 6 would be the dot-product-based similarity for a word pair.

$$\mathrm{SimMatrix} = M^{\mathcal{C}\to\mathcal{V}} M^{\mathcal{V}\to\mathcal{C}} \quad (6)$$

We introduce context weighting by inserting a square matrix W between the two (see Equation 7). Similarity is unchanged if W is the identity matrix. If W is a non-identity diagonal matrix, this is equivalent to treating some contexts as more important than others. It is by appropriately choosing weights in W that we achieve the context dependent similarity. If, for instance, all contexts other than those indicative of cars are zeroed out in W, *ford* and *obama* will have no similarity.

$$\mathrm{SimMatrix} = M^{\mathcal{C}\to\mathcal{V}} W M^{\mathcal{V}\to\mathcal{C}} \quad (7)$$

4.3 Set Expansion via Matrix Multiplication

To expand a set of k seeds, we can construct the k-hot column vector S with a 1 corresponding to each seed, and a 0 elsewhere. Given S, we calculate the focus matrix, W_S. Then the expansion E is a column vector that is just:

$$E = M^{\mathcal{C}\to\mathcal{V}} W_S M^{\mathcal{V}\to\mathcal{C}} S \quad (8)$$

The score for a term in E is the sum of its focused similarity to each seed.

4.4 Motivating Our Choice of W

When expanding the set {*taurus, cancer*}—the set of star signs, or perhaps the constellations—we are faced with the presence of a polysemous term with a lopsided polysemy. The *disease* sense is much more prevalent than the *star sign* sense for *cancer*, and the associated contexts are also unevenly distributed. If we attempt to use Equation 8 with the identity matrix W, the expansion is dominated by diseases.

The contexts we care about are those that are shared. Note that restricting ourselves to the intersection is not sensible, since if we are given a dozen seeds it is entirely possible that they share family resemblances and have a high pairwise overlap in contexts between any two seeds but where there are almost no contexts shared by all. We thus require a soft intersection, and this we achieve by downweighting contexts based on what fraction of the seeds are associated with that context. The parameter ρ described in the next section achieves this.

This modification helps, but it is not enough. Each disease-related context for *cancer* is now weakened, but their large number causes many diseases to rank high in the expansion. To address this, we can limit ourselves to only the top n contexts (typically, $n = 100$ is used). This way, if the joint contexts are highly ranked, the expansion will be based only on such contexts.

input : $S \subset \mathcal{V}$ (seeds), $\rho \in \mathbb{R}$ (limited support penalty), $n \in \mathbb{N}$ (context footprint)
output: The diagonal matrix W.

1 **for** $c \in \mathcal{C}$ **do**
2 | *// Activation of the context.*
3 | $a(c) \leftarrow \sum_{w \in S} M^{\mathcal{V}\to\mathcal{C}}_{w,c}$
4 | *// Fraction of S with context active*
5 | $f(c) \leftarrow$ fraction with $M^{\mathcal{V}\to\mathcal{C}}_{*,c} > 0$
6 | *// Score of context*
7 | $s(c) \leftarrow f(c)^\rho a(c)$
8 **end**
9 Sort contexts by score $s(c)$
10 **for** $c \in C$ **do**
11 | **if** *c one of n top-scoring contexts* **then**
12 | | $W_{c,c} = f(c)^\rho$
13 | **end**
14 **end**

Algorithm 1: Calculating context focus

The {*taurus, cancer*} example is useful to point out the benefits of an asymmetric association measure. Given *cancer*, the notion of *star sign* is not highly activated, and rightly so. If w is *cancer* and c is BORN UNDER X, then $\text{PPMI}(w, c)$ is low (as is $\text{APPMI}(w, c)$). However, $\text{APPMI}(c, w)$ is quite high, allowing us to highly score *cancer* when expanding {*taurus, aries*}.

4.5 Details of Calculating W

To produce W, we provide the seeds and two parameters: $\rho \in \mathbb{R}$ (the *limited support penalty*) and $n \in \mathbb{N}$ (the *context footprint*). Algorithm 1 provides the pseudo-code.

First, we score contexts by their *activation* (line 3). We penalize contexts that are not supported by all the seeds: we produce the score by multiplying activation by f^ρ, where f is the fraction of the seeds supporting that context (lines 5 and 7). Only the n top scoring contexts will have non-zero values in W, and these get the value f^ρ.

This notion of weighting contexts is similar to that used in the SetExpan framework (Shen et al., 2017), although the way they use it is different (they use weighted Jaccard similarity based on context weights). Their algorithm for calculating context weights is a special case of our algorithm, with no notion of *limited support penalty*, that is, they use $\rho = 0$.

4.6 Sparse Representations for Analogies

To solve the analogy problem "What is the Ganga of Egypt?" we are looking for something that is like *Ganga* (this we can obtain via the set expansion of the (singleton) set {*Ganga*}, as described above) and that we see often with *Egypt*, or to use Turney's terminology, in the same domain as *Egypt*.

To find terms that are in the same domain as a given term, we use the same statistical tools, merely with a different set of contexts. The context for a term is other terms in the same sentence. With this alternate definition of context, we produce $D^{\mathcal{C}\to\mathcal{V}}$ exactly analogous to $M^{\mathcal{C}\to\mathcal{V}}$ from Equation 5.

However, if we define $D^{\mathcal{V}\to\mathcal{C}}$ analogous to $M^{\mathcal{V}\to\mathcal{C}}$ and use these matrices for expansion, we run into unintended consequences since expanding {*evolution*} provides not what things *evolution* is seen with, but rather those things that cooccur with what *evolution* co-occurs with. Since, for example, both *evolution* and *number* co-occur with *theory*, the two would appear related. To get around this, we zero out most non-diagonal entries in $D^{\mathcal{V}\to\mathcal{C}}$. The only off diagonal entries that we do not zero out are those corresponding to word pairs that seem to share a lemma (which we heuristically define as "share more than 80% of the prefix". Future work will explore using lemmas). An example of a pair we retain is *india* and *indian*), since when we are looking for items that co-occur with *india* we actually want those that occur with related words forms. An illustration for why this matters: *India* and *Rupee* occur together rarely (with a negative PMI) whereas *Indian* and *Rupee* have a strong positive PMI.

4.7 Finding Analogies

To answer "What is the Ganga of Egypt", we use Equation 8 on the singleton set {*ganga*}, and the same equation (but with $D^{\mathcal{V}\to\mathcal{C}}$ and $D^{\mathcal{C}\to\mathcal{V}}$) on {*egypt*}. We intersect the two lists by combining the score of the shared terms in squash space (i.e., if the two scores are m and d, the combined score is

$$\frac{100m}{99+m} + \frac{100d}{99+d} \tag{9}$$

5 Set Expansion Experiments and Evaluation

5.1 Experimental Setup

We report data on two different corpora.

The Comparison Corpus. We begin with 20 million English web pages randomly sampled from a set of popular web pages (high pagerank according to Google). We run Word2Vec on the text of these pages, producing a 200 dimensional embeddings. We also produce $M^{\mathcal{V}\to\mathcal{C}}$ and $M^{\mathcal{C}\to\mathcal{V}}$ according to Equation 5. We use this corpus to compare Category Builder with Word2Vec-based techniques. Note that these web-pages may be noisier than Wikipedia. Word2Vec was chosen because it was deemed "comparable": mathematically, it is an implicit factorization of the PMI matrix (Levy and Goldberg, 2014b).

Release Corpus. We also ran Category Builder on a much larger corpus. The generated matrices are restricted to the most common words and phrases (around 200,000). The matrices and associated code are publicly available[1].

Using Word2Vec for Set Expansion. Two classes of techniques are considered, representing members of both families described by Gyllensten and Sahlgren (2018). The centroid method finds the centroid of the seeds and expands to its neighbors based on cosine similarity. The other methods first find similarity of a candidate to each seed, and combines these scores using arithmetic, geometric, and harmonic means.

Mean Average Precision (MAP). MAP combines both precision and recall into a single number. The gold data to evaluate against is presented as sets of synsets, e.g., $\{\{\textit{California}, \textit{CA}\}, \{\textit{Indiana}, \textit{IN}\}, \ldots\}$.

An expansion L consists of an ordered list of terms (which may include the seeds). Define $\text{Prec}_i(L)$ to be the fraction of items in the first i items in L that belong to at least one golden synset. We can also speak of the precision at a synset, $\text{Prec}_S(L) = \text{Prec}_j(L)$, where j is the smallest index where an element in S was seen in L. If no element in the synset S was ever seen, then $\text{Prec}_S = 0$. $\text{MAP}(L) = avg(\text{Prec}_S(L))$ is the average precision over all synsets.

Generalizations of MAP. While MAP is an excellent choice for closed sets (such as U.S. STATES), it is less applicable to open sets (say, POLITICAL IDEOLOGIES or SCIENTISTS). For such cases, we propose a generalization of MAP that preserves its attractive properties of combining precision and recall while accounting for variant names. The proposed score is $\text{MAP}_n(L)$, which is the average of precision for the first n synsets seen. That it is a strict generalization of MAP can be seen by observing that in the case of US STATES, $\text{MAP}(L) \equiv \text{MAP}_{50}(L)$.

[1] https://github.com/google/categorybuilder

5.2 Evaluation Sets

We produced three evaluation sets, two closed and one open. For closed sets, following Wang and Cohen (2007), we use US States and National Football League teams. To increase the difficulty, for NFL teams, we do not use as seeds dismabiguated names such as *Detroit Lions* or *Green Bay Packers*, instead using the polysemous *lions* and *packers*. The synsets were produced by adding all variant names for the teams. For example, *Atlanta Falcons* are also known as *falcs*, and so this was added to the synset.

For the open set, we use verbs that indicate things breaking or failing in some way. We chose ten popular instances (e.g., *break, chip, shatter*) and these act as seeds. We expanded the set by manual evaluation: any correct item produced by any of the evaluated systems was added to the list. There is an element of subjectivity here, and we therefore provide the lists used (Appendix A.1).

5.3 Evaluation

For each evaluation set, we did 50 set expansions, each starting with three randomly selected seeds.

Effect of ρ and APPMI. Table 1 reveals that APPMI performs better than PPMI — significantly better on two sets, and slightly worse on one. Penalizing contexts that are not shared by most seeds (i.e., using $\rho > 0$) also has a marked positive effect.

Effect of n. Table 2 reveals a curious effect. As we increase n, for US STATES, performance drops somewhat but for BREAK VERBS it improves quite a bit. Our analysis shows that pinning down what a state is can be done with very few contexts, and other shared contexts (such as LIVE IN X) are shared also with semantically related entities such as states in other countries. At the other end, BREAK VERBS is based on a large number of shared contexts and using more contexts is beneficial.

Technique	US States	NFL Teams	*Break* Verbs
W2V HM	.858	.528	.231
W2V GM	.864	.589	.273
W2V AM	.852	.653	.332
W2V Centroid	.851	.646	.337
CB:PPMI; $\rho = 0$	.918	.473	.248
CB:PPMI; $\rho = 3$	**.922**	.612	.393
CB:APPMI; $\rho = 0$	.900	.584	.402
CB:APPMI; $\rho = 3$	.907	**.735**	**.499**
CB:Release Data†	.959	.999	.797

Table 1: MAP scores on three categories. The first four rows use various techniques with Word2Vec. The next four demonstrate Category Builder built on the same corpus, to show the effect of ρ and association measure used. For all four Category Builder rows, we used $n = 100$. Both increasing ρ and switching to APPMI can be seen to be individually and jointly beneficial. †The last line reports the score on a different corpus, the release data, with APPMI and $\rho = 3, n = 100$.

	5	10	30	50	100	500
US States	**.932**	.925	.907	.909	.907	.903
NFL	.699	.726	.731	.734	**.735**	.733
Break Verbs	.339	.407	.477	.485	.496	**.511**

Table 2: Effect of varying n. APPMI with $\rho = 3$.

5.4 Error Analysis.

Table 3 shows the top errors in expansion. The kinds of drifts seen in the two cases are revealing. Category Builder picks up word fragments (e.g., because of the US State *New Mexico*, it expanded states to include *Mexico*). It sometimes expands to a hypernym (e.g., *province*) or siblings (e.g., instead of Football teams sometimes it got other sport teams). With Word2Vec, we see similar errors (such as expanding to the semantically similar *southern california*).

5.5 Qualitative Demonstration

Table 4 shows a few examples of expanding categories, with $\rho = 3, n = 100$.

Table 5 illustrates the power of Category Builder by considering a a synthetic corpus produced by replacing all instances of *cat* and *denver* into the hypothetical *CatDenver*. This illustrates that even without explicit WSD (that is, separating *CatDenver* to its two "senses", we are able to expand correctly given an appropriate context. To complete the picture, we note that expanding {*kitten, dog*} as well as {*atlanta, phoenix*} contains *CatDenver*, as expected.

Set	Method	Top Errors
US States	W2V	southern california; east tennessee; seattle washington
	CB	carolina; hampshire; dakota; ontario; jersey; province
NFL	W2V	hawks; pelicans; tigers; nfl; quarterbacks; sooners
	CB	yankees; sox; braves; mets; knicks; rangers, lakers

Table 3: Error analysis for US States and NFL. Arithmetic Mean method is used for W2V and $\rho = 3$ and APPMI for Category Builder

Seeds	**CB Expansion,** $\rho = 3, n = 100$
ford, nixon	nixon, ford, obama, clinton, bush, richard nixon, reagan, roosevelt, barack obama, bill clinton, ronald reagan, w. bush, eisenhower
ford, chevy	ford, chevy, chevrolet, toyota, honda, nissan, bmw, hyundai, volkswagen, audi, chrysler, mazda, volvo, gm, kia, subaru, cadillac
ford, depp	ford, depp, johnny depp, harrison ford, dicaprio, tom cruise, pitt, khan, brad pitt, hanks, tom hanks, leonardo dicaprio
safari, trip†	trip, safari, tour, trips, cruise, adventure, excursion, vacation, holiday, road trip, expedition, trek, tours, safaris, journey,
safari, ie†	safari, ie, firefox, internet explorer, chrome, explorer, browsers, google chrome, web browser, browser, mozilla firefox

Table 4: Expansion examples using Category Builder so as to illustrate its ability to deal with Polysemy. † For these examples, $\rho = 5$

6 Analogies

6.1 Experimental Setup

We evaluated the analogy examples used by Mikolov et al. (2013). Category Builder evaluation were done by expanding using syntactic and sentence-based-cooccurrence contexts as detailed in Section 4.6 and scoring items according to Equation 9. For evaluating using Word2Vec, the standard vector arithmetic was used.

In both cases, the input terms from the problem were removed from candidate answers (as was done in the original paper). Linzen (2016) provides analysis and rationales for why this is done.

6.2 Evaluation

Table 6 provides the evaluations. A few words are in order for the difference between the published scores for Word2Vec analogies elsewhere (e.g., Linzen, 2016). Their reported numbers for common capitals were around 91%, as opposed to around 87% here. Where as Wikipedia is typically used as a corpus, that was not the case here. Our corpus is noisier, and may not have the same level

Seeds	CB Expansion, $\rho = 3, n = 100$
CatDenver, dog	dogs, cats, puppy, pet, rabbit, kitten, animal, animals, pup, pets, puppies, horse
CatDenver, phoenix	chicago, atlanta, seattle, dallas, boston, portland, angeles, los angeles
CatDenver, TigerAndroid	cats, lion, dog, tigers, kitten, animal, dragon, wolf, dogs, bear, leopard, rabbit

Table 5: Expansion examples with synthetic polysemy by replacing all instances of *cat* and *denver* into the hypothetical *CatDenver* (similarly, *TigerAndroid*). A single other term is enough to pick out the right sense.

	a:*b*::*c*:*?*	Harder :*b*::*c*:*?*(*a* withheld)	
Method	**W2V**	**CB:APPMI**	
Corpus	Comp	Comp	**Release**†
common capitals	.872	**.957**	.941
city-in-state	.657	**.972**	.955
currency	.030	**.037**	.122
nationality	.515	**.615**	.655
world capitals	.472	**.789**	.668
family	**.617**	.217	.306

Table 6: Performance on Analogy classes from Mikolov et al. (2013). The first two columns are derived from the same corpus, whereas the last column reports numbers on the data we will release. For category builder, we used $\rho = 3, n = 100$

of country-based factual coverage as Wikipedia, and almost all non-grammar based analogy problems are of that nature.

A second matter to point out is why grammar based rows are missing from Table 6. Grammar based analogy classes cannot be solved with just two terms. For *boy*:*boys*::*king*:*?*, dropping the first term *boy* takes away information crucial to the solution in a way that dropping the first term of *US*:*dollar*::*India*:*?* does not. The same is true for the *family* class of analogies.

6.3 Qualitative Demonstration

Table 7 provides a sampler of analogies solved using Category Builder.

7 Limitations

Much work remains, of course. The analogy work presented here (and also the corresponding work using vector offsets) is no match for the subtlety that people can bring to bear when they see deep connections via analogy. Some progress here could come from the ability to discover and use more semantically meaningful contexts.

There is currently no mechanism to automatically choose n and ρ. Standard settings of $n = 100$ and $\rho = 3$ work well for the many applications we use it for, but clearly there are categories that benefit from very small n (such as BLUE THINGS) or very large n. Similarly, as can be seen in Equation 9, analogy also uses a parameter for combining the results, with no automated way yet to choose it. Future work will prioritize this.

$term_1$	$term_2$	**What is the $term_1$ of $term_2$?**
voldemort	tolkien	sauron
voldemort	star wars	vader
ganga	egypt	nile
dollar	india	rupee
football	india	cricket
civic	toyota	corolla

Table 7: A sampler of analogies solved by Category Builder.

The current work suggests, we believe, that it is beneficial to not collapse the large dimensional sparse vector space that implicitly underlies many embeddings. Having the ability to separately manipulate contexts can help differentiate between items that differ on that context. That said, the smoothing and generalization that dimensionality reduction provides has its uses, so finding a combined solution might be best.

8 Conclusions

Given that natural categories vary in their degree of similarities and their kinds of coherence, we believe that solutions that can adapt to these would perform better than context independent notions of similarity.

As we have shown, Category Builder displays the ability to implicitly deal with polysemy and determine similarity in a context sensitive manner, as exhibited in its ability to expand a set by latching on to what is common among the seeds.

In developing it we proposed a new measure of association between words and contexts and demonstrated its utility in set expansion and a hard version of the analogy problem. In particular, our results show that sparse representations deserve additional careful study.

Acknowledgments

We are thankful to all the reviewers for their helpful comments and critiques. In particular, Ido Dagan, Yoav Goldberg, Omer Levy, Praveen Paritosh, and Chris Waterson gave us insightful comments on earlier versions of this write up. The research of Dan Roth is partly supported by a Google gift and by DARPA, under agreement number FA8750-13-2-008.

References

Michael Agar. 1994. *Language shock: Understanding the culture of conversation*. William Morrow & Company.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Linear algebraic structure of word senses, with applications to polysemy. *arXiv preprint arXiv:1601.03764*.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 232–239, Prague, Czech Republic. Association for Computational Linguistics.

John M. Ellis. 1993. *Language, Thought, and Logic*. Northwestern University Press.

A. R. Golding and D. Roth. 1999. A winnow based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

Robert L Goldstone, Yvonne Lippa, and Richard M Shiffrin. 2001. Altering object representations through category learning. *Cognition*, 78(1):27–43.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2018. Distributional term set expansion.

Douglas Hofstadter and Emmanuel Sander. 2013. *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. Basic Books.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

P. Jindal and D. Roth. 2011. Learning from negative examples in set-expansion. In *ICDM*, pages 1110–1115.

George Lakoff. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. University of Chicago Press.

Omer Levy. 2018. *The Oxford handbook of computational linguistics*, second edition, chapter Word Representaion. Oxford University Press.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding?

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. *arXiv preprint arXiv:1606.07736*.

Douglas L Medin, Robert L Goldstone, and Dedre Gentner. 1993. Respects for similarity. *Psychological review*, 100(2):254.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

James Pustejovsky and Bran Boguraev. 1997. *Lexical semantics: The problem of polysemy*. Clarendon Press.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Francois Role and Mohamed Nadif. 2011. Handling the impact of low frequency events on co-occurrence based measures of word similarity. In *International Conference on Knowledge Discovery and Information Retrieval*, pages 226–231.

Xin Rong, Zhe Chen, Qiaozhu Mei, and Eytan Adar. 2016. Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 645–654. ACM.

A. Rozovskaya and D. Roth. 2014. Building a state-of-the-art grammatical error correction system.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 288–304. Springer.

Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Richard C Wang and William W Cohen. 2007. Language-independent set expansion of named entities using the web. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 342–350. IEEE.

A Supplemental Material

A.1 Lists Used in Evaluating Set Expansion

US States. Any of the 50 states could be used as a seed. The 50 golden synsets were the 50 pairs of state name and abbreviation (e.g., {*California, CA*}).

NFL Teams. Any of the first terms in these 32 synsets could be used as a seed. The golden synsets are: {*Bills, Buffalo Bills*}, {*Dolphins, Miami Dolphins, Phins*}, {*Patriots, New England Patriots, Pats*}, {*Jets, New York Jets*}, {*Ravens, Baltimore Ravens*}, {*Bengals, Cincinnati Bengals*}, {*Browns, Cleveland Browns*}, {*Steelers, Pittsburgh Steelers*}, {*Texans, Houston Texans*}, {*Colts, Indianapolis Colts*}, {*Jaguars, Jacksonville Jaguars, Jags*}, {*Titans, Tennessee Titans*}, {*Broncos, Denver Broncos*}, {*Chiefs, Kansas City Chiefs*}, {*Chargers, Los Angeles Chargers*}, {*Raiders, Oakland Raiders*}, {*Cowboys, Dallas Cowboys*}, {*Giants, New York Giants*}, {*Eagles, Philadelphia Eagles*}, {*Redskins, Washington Redskins*}, {*Bears, Chicago Bears*}, {*Lions, Detroit Lions*}, {*Packers, Green Bay Packers*}, {*Vikings, Minnesota Vikings, Vikes*}, {*Falcons, Atlanta Falcons, Falcs*}, {*Panthers, Carolina Panthers*}, {*Saints, New Orleans Saints*}, {*Buccaneers, Tampa Bay Buccaneers, Bucs*}, {*Cardinals, Arizona Cardinals*}, {*Rams, Los Angeles Rams*}, {*49ers, San Francisco 49ers, Niners*}, and {*Seahawks, Seattle Seahawks*}

Break Verbs. Seeds are chosen from among these ten items: *break, chip, shatter, rot, melt, scratch, crush, smash, rip, fade*. Evaluation is done for MAP_{30} (see Section 5.1). The following items are accepted in the expansion: *break up, break down, tip over, splinter, tear, come off, crack, disintegrate, deform, crumble, burn, dissolve, bend, chop, stain, destroy, smudge, tarnish, explode, derail, deflate, corrode, trample, ruin, suffocate, obliterate, topple, scorch, crumple, pulverize, fall off, cut, dry out, split, deteriorate, hit, blow, damage, wear out, peel, warp, shrink, evaporate, implode, scrape, sink, harden, abrade, unhinge, erode, calcify, vaporize, sag, shred, degrade, collapse, annihilate*. In the synsets, we also added the morphological variants (e.g., {*break, breaking, broke, breaks*}).

A.2 Word2Vec Model Details

The word2vec model on the "comparison corpus" created 200 dimensional word embeddings. We used a skip-gram model with a batch size of 100, a vocabulary of 600k ngrams, and negative sampling with 100 examples. It was trained using a learning rate of 0.2 with Adagrad optimizer for 70 million steps.