The (Too Many) Problems of Analogical Reasoning with Word Vectors

Anna Rogers
Dept. of Computer Science
University of Massachusetts Lowell
Lowell, MA, USA
arogers@cs.uml.edu

Aleksandr Drozd
Global Scient. Inf. and Comput. Center
Tokyo Institute of Technology
Tokyo, Japan
alex@smg.is.titech.ac.jp

Bofang Li
School of Information
Renmin University of China
Beijing, China
libofang@ruc.edu.cn

Abstract

This paper explores the possibilities of analogical reasoning with vector space models. Given two pairs of words with the same relation (e.g. *man:woman :: king:queen*), it was proposed that the offset between one pair of the corresponding word vectors can be used to identify the unknown member of the other pair ($\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = ?\overrightarrow{queen}$). We argue against such "linguistic regularities" as a model for linguistic relations in vector space models and as a benchmark, and we show that the vector offset (as well as two other, better-performing methods) suffers from dependence on vector similarity.

1 Introduction

This paper considers the phenomenon of "vector-oriented reasoning" via linear vector offset in vector space models (VSMs) (Mikolov et al., 2013c,a). Given two pairs of words with the same linguistic relation (*woman:man :: king:queen*), it has been proposed that the offset between one pair of word vectors can be used to identify the unknown member of a different pair of words via solving proportional analogy problems ($\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = ?\overrightarrow{queen}$), as shown in Fig. 1. We will refer to this method as 3CosAdd.

This approach attracted a lot of attention, both as the "poster child" of word embeddings, and for its potential practical utility. Given the vital role that analogical reasoning plays in human cognition for discovering new knowledge and understanding new concepts, automated analogical reasoning could become a game-changer in many fields, providing a universal mechanism for detecting linguistic relations (Turney, 2008) and word sense disambiguation (Federici et al., 1997). It is already used in many downstream NLP tasks, such as splitting compounds (Daiber et al., 2015), semantic search (Cohen et al., 2015), cross-language relational search (Duc et al., 2012), to name a few.

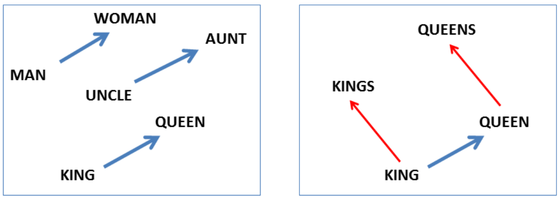


Figure 1: Linguistic relations modeled by linear vector offset (Mikolov et al., 2013c)

The idea that linguistic relations are mirrored in neat geometrical relations (as shown in Fig. 1) is also intuitively appealing, and 3CosAdd has become a popular benchmark. Roughly, the current VSMs score between 40% (Lai et al., 2016) and 75% (Pennington et al., 2014) on the Google test set (Mikolov et al., 2013a). However, in fact performance varies widely for different types of relations (Levy and Goldberg, 2014; Köper et al., 2015; Gladkova et al., 2016).

One way to explain the current limitations is to attribute them to the imperfections of the current models and/or corpora with which they are built: with this view, in a perfect VSM, any linguistic relation should be recoverable via vector offset.

The alternative to be explored in this paper is that perhaps natural language semantics is more complex than suggested by Fig. 1, and there may be both theoretical and mathematical issues with analogical reasoning with word vectors and its 3CosAdd implementation.

We present a series of experiments with two popular VSMs (GloVe and Word2Vec) to show that the accuracy of 3CosAdd depends on the proximity of the target vector to its source (i.e.

*Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148,
Vancouver, Canada, August 3-4, 2017. ©2017 Association for Computational Linguistics

$\overrightarrow{queen}$ should be quite similar to $\overrightarrow{king}$). Since not all linguistic relations can be expected to result in high word vector proximity, the method is limited to those that happen to be so in a given VSM. Furthermore, its accuracy also varies because the "linguistic regularities" are actually not so regular, and should not be expected to be so. We also compare 3CosAdd to two alternative methods to investigate whether better algorithms can improve on these and other accounts.

2 Background: "Relational Similarity" vs "Word Analogies"

The most fundamental term for what 3CosAdd is supposed to capture is actually not analogy, but rather *relational similarity*, i.e. the idea that pairs of words may hold similar relations to those between other pairs of words. For example, the relation between *cat* and *feline* is similar to the relation between *dog* and *canine*. Notably, this is *similarity* rather than identity: "instances of a single relation may still have significant variability in how characteristic they are of that class" (Jurgens et al., 2012).

Analogy as it is known in philosophy and logic is something quite different. The "classical" analogical reasoning follows roughly this template: objects X and Y share properties a, b, and c; therefore, they may also share the property d. For example, both Earth and Mars orbit the Sun, have at least one moon, revolve on axis, and are subject to gravity; therefore, if Earth supports life, so could Mars (Bartha, 2016).

The NLP move from relational similarity to analogy follows the use of the term by P. Turney, who distinguishes between attributional similarity between two words and relational similarity between two pairs of words. On this interpretation, *two word pairs that have a high degree of relational similarity are analogous* (Turney, 2006).

In terms of practical NLP tasks, Turney et al. (2003) introduced the task of solving SAT[1] analogy problems by choosing from several provided options. These problems were formulated as *proportional analogies*, written in the form $a : a' :: b : b'$ (a is to a' as b is to b')

It is this use of the term "analogy" that Mikolov et al. (2013c) followed in proposing the 3CosAdd method. They formulated the task as selecting a single best fitting vector out of the whole vocabulary of the VSM. It became known as *word analogy task*, but in its core it is still basically estimation of relational similarity, and could be formulated as such: given a pair of words a and a', find how they are related and then find word b', such that it has a similar relation with the word b. A crucial difference is that the graded, non-binary nature of relational similarity is now not in focus: the goal is to find a single correct answer.

The dataset that came to be known as the Google analogy test set (Mikolov et al., 2013a), included 14 linguistic relations with 19544 questions in total. It has become one of the most popular benchmarks for VSMs. This evaluation paradigm assumes that:

(1) Words in similar linguistic relations should in principle be recoverable via relational similarity to known word pairs.

(2) 3CosAdd score reflects the extent to which a given VSM encodes linguistic relations.

(1) became dubious when it was shown that accuracy of 3CosAdd varies widely between categories (Levy and Goldberg, 2014), and even the best-performing GloVe model scores under 30% on the more challenging Bigger Analogy Test Set (BATS) (Gladkova et al., 2016). It appears that not all relations can be identified in this way, with lexical semantic relations such as synonymy and antonymy being particularly difficult (Köper et al., 2015; Vylomova et al., 2016). The assumption of a single best-fitting candidate answer is also being targeted (Newman-Griffis et al., 2017).

(2) was refuted when Drozd et al. (2016) demonstrated that some relations missed by 3CosAdd could be recovered with a supervised method, and therefore the information was present in the VSM – just not recoverable with 3CosAdd.

Let us consider why both (1) and (2) failed.

3 What Does 3CosAdd Really Do?

3.1 Methodology

We present a series of experiments performed with BATS dataset. Although there are more results on analogy task published with Google test than with BATS, Google test only contains 15 types of linguistic relations, and these happen to be the easier ones (Gladkova et al., 2016).

Table 1 lists examples of each BATS category: there are 50 word pairs for each of 40 linguistic

[1] Scholastic Aptitude Test.

Inflectional morphology	Nouns	regular plurals *(student:students)*, plurals with orthographic changes *(wife:wives)*
	Adjectives	comparative degree *(strong:stronger)*, superlative degree *(strong:strongest)*
	Verbs	infinitive: 3Ps.Sg *(follow:follows)*, infinitive: participle *(follow:following)*, infinitive: past *(follow:followed)*, participle: 3Ps.Sg *(following:follows)*, participle: past *(following:followed)*, 3Ps.Sg : past *(follows:followed)*
Derivational morphology	Stem change	verb+er *(bake:baker)*, verb+able *(edit:editable)*, verb+ation *(continue:continuation)*, verb+ment *(argue:argument)*
	No stem change	re+verb *(create:recreate)*, noun+less *(home:homeless)*, adj.+ness *(mad:madness)*, un+adj. *(able:unable)*, adj.+ly *(usual:usually)*, over+adj. *(used:overused)*
Lexicographic semantics	Hypernyms	animals *(turtle:reptile)*, miscellaneous *(peach:fruit)*
	Hyponyms	miscellaneous *(color:white)*
	Meronyms	part-whole *(car:engine)*, substance *(sea:water)*, member *(player:team)*,
	Antonyms	opposites *(up:down)*, gradable *(clean:dirty)*
	Synonyms	exact *(sofa:couch)*, intensity *(cry:scream)*
Encyclopedic semantics	Animals	the young *(cat:kitten)*, sounds *(dog:bark)*, shelter *fox:den*
	Geography	capitals *(Athens:Greece)*, languages *(Peru:Spanish)*, UK city:county *York:Yorkshire*
	People	occupation *(Lincoln:president)*, nationalities *(Lincoln:American)*
	Other	thing:color *(blood:red)*, male:female *(actor:actress)*

Table 1: The Bigger Analogy Test Set: categories and examples

relations (98,000 questions in total). BATS covers most relations in the Google set, but it adds many new and more difficult relations, balanced across derivational and inflectional morphology, lexicographic and encyclopedic semantics (10 relations of each type). Thus BATS provides a less flattering, but more accurate estimate of the capacity for analogical reasoning in the current VSMs.

We use pre-trained GloVe vectors by Pennington et al. (2014), released by the authors[2] and trained on Gigaword 5 + Wikipedia 2014 (300 dimensions, window size 10). We also experiment with Word2Vec vectors (Mikolov et al., 2013b) released by the authors[3], trained on a subcorpus of Google news (also with 300 dimensions).

The evaluation with 3CosAdd and LRCos methods was conducted with the Python script that accompanies BATS. We also added an implementation of 3CosMul, a multiplicative objective proposed by Levy and Goldberg (2014), now available in the same script[4]. Since 3CosMul requires normalization, we used normalized GloVe and Word2Vec vectors in all experiments.

Questions with words not in the model vocabulary were excluded (0.01% BATS questions for GloVe and 0.016% for Word2Vec).

3.2 The "Honest" 3CosAdd

Let us remember that 3CosAdd as initially formulated by Mikolov et al. (2013c) excludes the three source vectors a, a' and b from the pool of possible answers. Linzen (2016) showed that if that is not done, the accuracy drops dramatically, hitting zero for 9 out of 15 Google test categories.

Let us investigate what happens on BATS data, split by 4 relation types. The rows of Fig. 2 represent all questions of a given category, with darker color indicating higher percentage of predicted vectors being the closest to a, a', b, b', or any other vector.

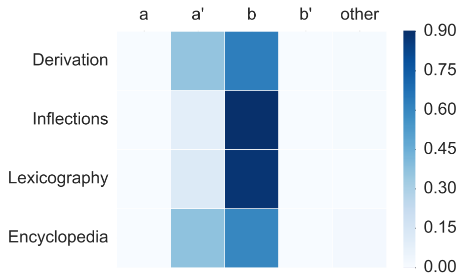


Figure 2: The result of $a - a' + b$ calculation on BATS: source vectors a, a', and b are not excluded.

Fig. 2 shows that if we do not exclude the source vectors, b is the most likely to be predicted; in derivational and encyclopedic categories a' is also possible in under 30% of cases. b' is as unlikely to be predicted as a, or any other vector.

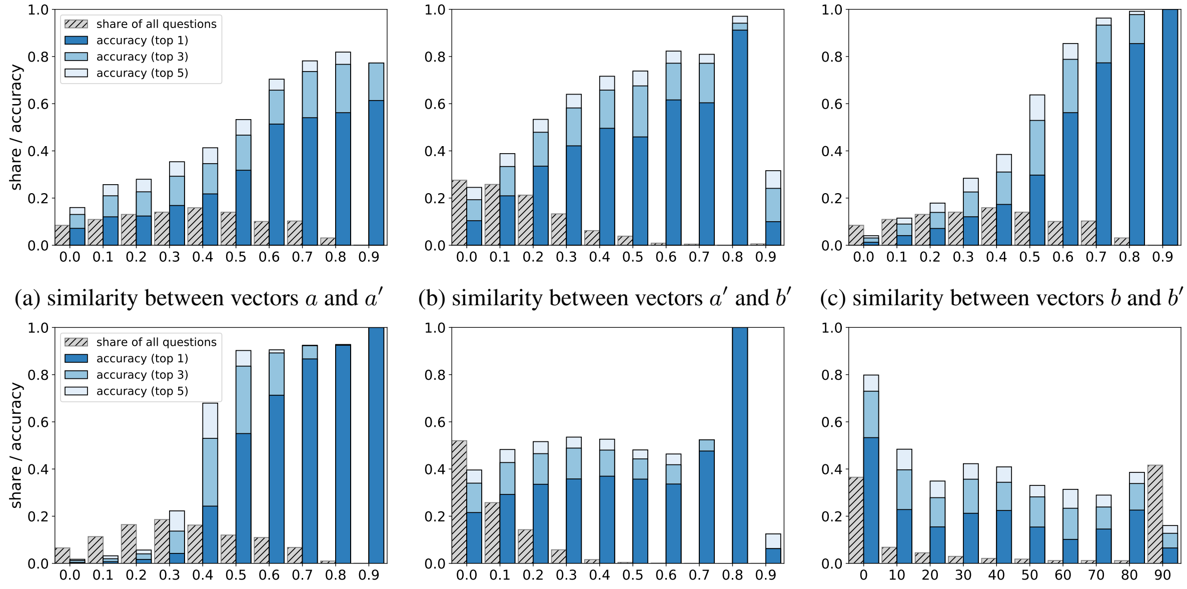
This experiment suggests that the addition of the offset between a and a' typically has a very small effect on the b vector – not sufficient to induce a shift to a different vector on its own. This would in effect limit the search space of 3CosAdd to the close neighborhood of the b vector.

It explains another phenomenon pointed out by Linzen (2016): for the plural noun category in the

[2] https://nlp.stanford.edu/projects/glove/

[3] https://code.google.com/archive/p/word2vec/

[4] http://vsm.blackbird.pw/tools/



(a) similarity between vectors a and a'

(b) similarity between vectors a' and b'

(c) similarity between vectors b and b'

(d) similarity between vector b' and predicted vector

(e) similarity between vector b' and a

(f) rank of b in the neighborhood of b'

*X-axis labels indicate lower boundary of the corresponding similarity/rank bins.
The numerical values for all data can be found in the Appendix.

Figure 3: Accuracy of 3CosAdd method on GloVe vs characteristics of the vector space.

Google test set 70% accuracy was achieved by simply taking the closest neighbor of the vector b, while 3CosAdd improved the accuracy by only 10%. That would indeed be expected if most singular (a) and plural (a') forms of the same noun were so similar, that subtracting them would result in a nearly-null vector which would not change much when added to b.

3.3 Distance to the Target Vector

Levy and Goldberg (2014, p.173) suggested that 3CosAdd method is "mathematically equivalent to seeking a word (b') which is similar to b and a' but is different from a." We examined the similarity between all source vector pairs, looking not only at the actual, top-1 accuracy of the 3CosAdd (i.e. the vector the closest to the hypothetical vector), but also at whether the correct answer was found in the top-3 and top-5 neighbors of the predicted vector. For each similarity bin we also estimated how many questions of the whole BATS dataset there were. The results are presented in Fig. 3.

Our data indicates that, indeed, for all combinations of source vectors, the accuracy of 3CosAdd decreases as their distance in vector space increases. It is the most successful when all three source vectors are relatively close to each other and the target vector. This is in line with the above evidence from the "honest" 3CosAdd: if the offset is typically small, for it to lead to the target vector, that target vector should be close.

Consider also the ranks of the b vectors in the neighborhood of b', shown in Fig. 3f. For nearly 40% of the successful questions b' was within 10 neighbors of b – and over 40% of low-accuracy questions were over 90 neighbors away.

As predicted by Levy et al., b' and a vectors do not exhibit the same clear trend for higher accuracy with higher similarity that is observed in all other cases (Fig. 3f). However, in experiments with only 20 morphological categories we did observe the same trend for b' and a as for the other vector pairs (see Fig. 4). This is counter-intuitive, and requires further examination.

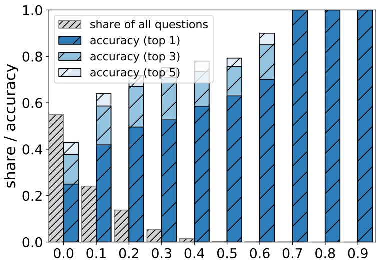


Figure 4: The similarity between b' and a on GloVe: morphological BATS categories only.

The observed correlation between the accuracy of 3CosAdd and the distance to the target vector could explain in particular the overall lower performance on BATS derivational morphology questions (only 0.08% top-1 accuracy) as opposed to inflectional (0.59%) or encyclopedic semantics (0.26%). $\overrightarrow{man}$ and $\overrightarrow{woman}$ could be expected to be reasonably similar distributionally, as they combine with many of the same verbs: both men and women sit, sleep, drink etc. However, the same could not be said of words derived with prefixes that change part of speech. Going from $\overrightarrow{happy}$ to $\overrightarrow{happiness}$, or from $\overrightarrow{govern}$ to $\overrightarrow{government}$, is likely to have to take us further in the vector space.

To make sure that the above trend is not specific to GloVe, we repeated these experiments with Word2Vec, which exhibited the same trends. All data is presented in Appendix A.1.

3.4 Uniqueness of a Relation

Note that the dependence of 3CosAdd on similarity is not entirely straightforward: Fig. 3b shows that for the highest similarity (0.9 and more) there is actually a drop in accuracy. The same trend was observed with Word2Vec (Fig 10 in Appendix 1). Theoretically, it could be attributed to there not being much data in the highest similarity range; but BATS has 98,000 questions, and even 0.1% of that is considerable.

The culprit is the "dishonesty" of 3CosAdd: as discussed above, it excludes the source vectors a, a', and b from the pool of possible answers. Not only does this mask the real extent of the difference between a and a', but it also creates a fundamental difficulty with categories where the source vectors may be the correct answers.

This is what explains the unexpected drops in accuracy at the highest similarity between vectors b' and a'. Consider the question $\overrightarrow{blood}$:$\overrightarrow{red}$ :: $\overrightarrow{snow}$:?$\overrightarrow{white}$. The vector offset could theoretically solve it, but if the question is $\overrightarrow{snow}$:$\overrightarrow{white}$ :: $\overrightarrow{sugar}$:?$\overrightarrow{white}$, the correct answer would *a priori* be excluded. In BATS data, this factor affects several semantic categories, including country:language, thing:color, animal:young, and animal:shelter.

3.5 Density of Vector Neighborhoods

If solving proportional analogies with word vectors is like shooting, the farther away the target vector is, the more difficult it should be to hit. Also, we can hypothesize that the more crowded a particular region is, the more difficult it should be to hit a particular target.

However, density of vector neighborhoods is not as straightforward to measure as vector similarity. We could look at average similarity between, e.g., top-10 ranking neighbors, but that could misrepresent the situation if some neighbors were very close and some were very far.

In this experiment we estimate density as the similarity to the 5th neighbor. The higher it is, the more highly similar neighbors a word vector has. This approach is shown in Fig. 5.

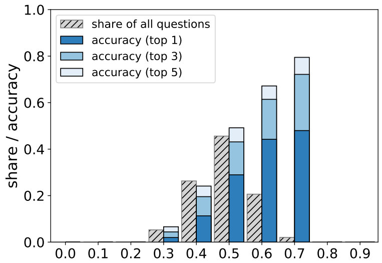


Figure 5: The similarity between b' and its 5th neighbor

The results seem counter-intuitive: denser neighborhoods actually yield higher accuracy (although there are virtually no cases of very tight neighborhoods). One explanation could be its reverse correlation with distance: if the neighborhood of b' is sparse, the closest word is likely to be relatively far away. But that runs contrary to the above findings that closer source vectors improve the accuracy of 3CosAdd. Then we could expect lower accuracy in sparser neighborhoods.

In this respect, too, GloVe and Word2Vec behave similarly (Fig. 15).

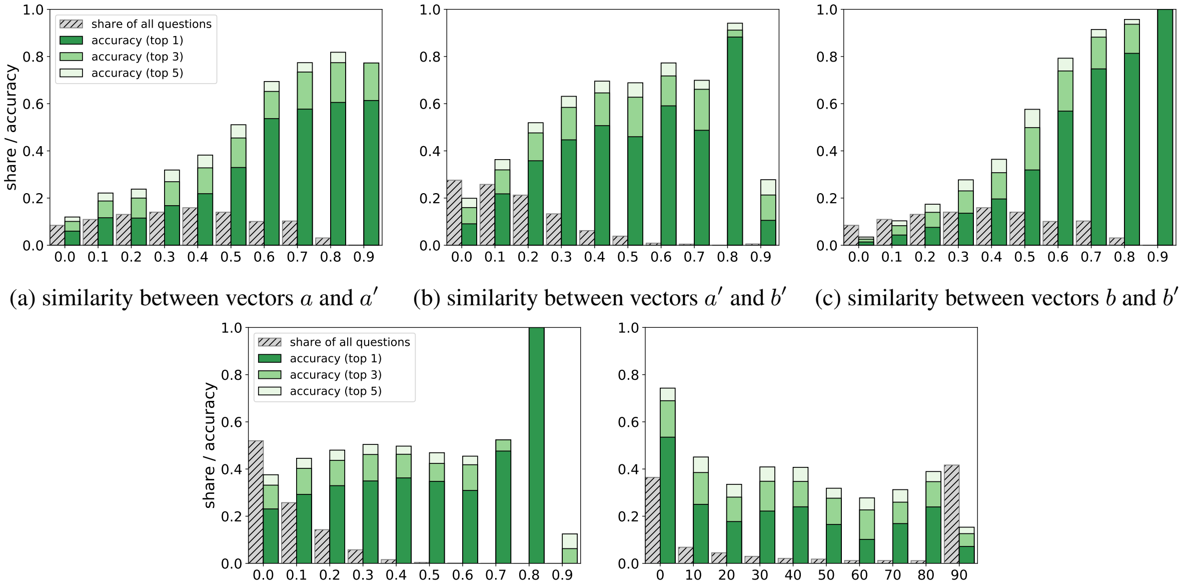
4 Comparison with Other Methods

We repeat the above experiments on GloVe with 3CosMul, a multiplication-based alternative to 3CosAdd proposed by Levy and Goldberg (2014):

$$argmax_{b' \in V} \frac{cos(b', b) cos(b', a')}{cos(b', a) + \varepsilon}$$

($\varepsilon = 0.001$ is used to prevent division by zero)

As 3CosMul does not explicitly calculate the predicted vector, we did not plot the similarity of b' to the predicted vector. But for other vector pairs shown in Fig. 6, we can see that 3CosMul,



(a) similarity between vectors a and a'

(b) similarity between vectors a' and b'

(c) similarity between vectors b and b'

(d) similarity between vector b' and a

(e) rank of b in the neighborhood of b'

*X-axis labels indicate lower boundary of the corresponding similarity/rank bins.
The numerical values for all data can be found in the Appendix.

Figure 6: Accuracy of 3CosMul method on GloVe model vs characteristics of the vector space.

like 3CosAdd, has much higher chances of success where target vectors are close to the source.

We also consider LRCos, a method based on supervised learning from a set of word pairs (Drozd et al., 2016). LRCos reinterprets the analogy task as follows: given a set of word pairs (e.g. *brother:sister, husband:wife, man:woman*, etc.), the available examples of the class of the target b' vector (*sister, wife, woman*, etc.) and randomly selected negative examples are used to learn a representation of the target class with a supervised classifier. The question is this: what word is the closest to $\overrightarrow{king}$, but belongs to the "women" class?

With LRCos it is only meaningful to look at the similarity of b to b' (Fig. 7). Once again, we see the same trend: closer targets are easier to hit.

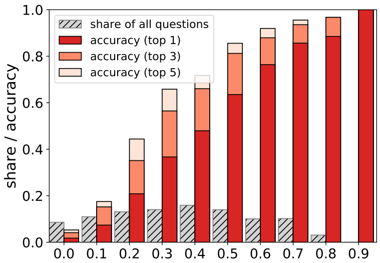


Figure 7: Accuracy of LRCos method vs similarity between vectors b and b'

However, if we look at overall accuracy, there is a big difference between the three methods.

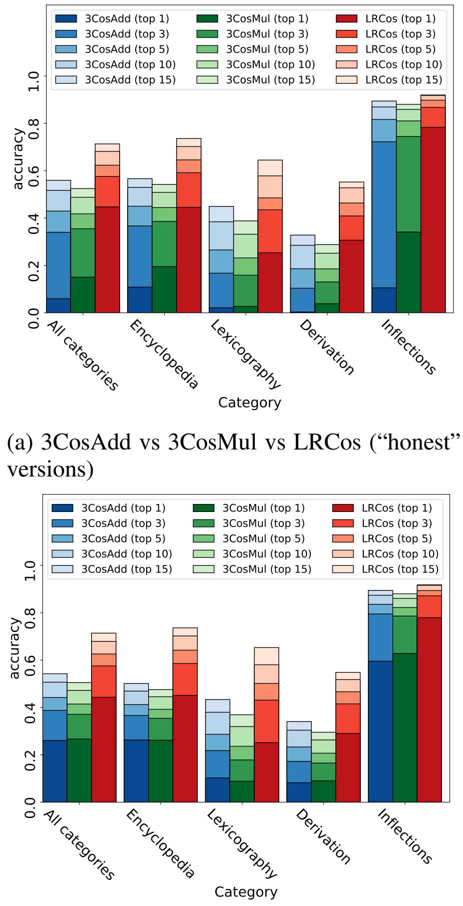
Fig. 8b shows that the accuracy of LRCos is much higher than the top-1 3CosAdd or 3CosMul. Moreover, its "honest" version (Fig. 8a) performs just as well as the "dishonest" one. These results are consistent with the results reported by Drozd et al. (2016). As for 3CosMul, Levy et al. (2015) show that 3CosMul outperforms 3CosAdd in PPMI, SGNS, GloVe and SVD models with the Google dataset, sometimes yielding 10-25% improvement. Our BATS experiment confirms the overall superiority of 3CosMul to 3CosAdd, although the difference is less dramatic.

Thus LRCos considerably outdoes its competitors, although it does not manage to avoid the similarity problem. We attribute this to the set-based, supervised nature of LRCos that gives it an edge on a different problem that affects both 3CosAdd and 3CosMul: the assumption of "linguistic regularities" from which we started.

5 Discussion: What Should We Expect from the Word Analogy Task?

5.1 How Regular Are "Linguistic Regularities"?

There are unresolved questions about the underlying assumption that the offset between vectors a'



(a) 3CosAdd vs 3CosMul vs LRCos ("honest" versions)

(b) CosAdd vs 3CosMul vs LRCos

Figure 8: LRCos performance on BATS

and a provides access to certain features combinable with vector b to detect b', and that such offset should be more or less constant for all words in a given linguistic relations.

Table 2 shows that this does not happen in a reliable way (data: BATS category D06 "re+verb").

Table 2: 3CosAdd: effect of various a : a' vector pairs with the same b : b' pair ($\overrightarrow{marry}$:$\overrightarrow{remarry}$)

No	a	a'	b	predicted vector	Sim. score	correct b' score
1	acquire	reacquire	marry	fiancée	0.54	<0.51
2	tell	retell	marry	betrothed	0.51	0.49
3	engage	reengage	marry	eloped	0.52	0.51
4	appear	reappear	marry	marries	0.65	0.55
5	establish	reestablish	marry	marries	0.58	0.52
6	invest	reinvest	marry	marries	0.59	0.57
7	adjust	readjust	marry	marrying	0.59	0.55
8	arrange	rearrange	marry	marrying	0.52	0.43
9	discover	rediscover	marry	marrying	0.54	0.49
10	apply	reapply	marry	remarry	0.53	0.53

Both correct and incorrect answers lie in about the same similarity range, so we cannot attribute the failures to the reliance of 3CosAdd on close neighborhoods. The distance from $\overrightarrow{marry}$ to $\overrightarrow{remarry}$ is the same; thus it must be the case that the offset between different a and a' is not the same, and leads to different answers – with a frustratingly small margin of error.

5.2 Can We Just Blame the Corpus?

Source corpora are noisy, and it is tempting to blame almost anything on that. It could be literal text-processing noise (e.g. not quite cleaned HTML data and ad texts) or, more broadly, any kind of information in the VSM that is irrelevant to the question at hand. This includes polysemy: for a word-level VSM the difference between $\overrightarrow{king}$ and $\overrightarrow{queen}$ is not exactly the same as the difference between $\overrightarrow{man}$ and $\overrightarrow{woman}$ just for the existence of the *Queen* band (although that factor should not affect the "re-" prefix verbs in Table 2).

In addition to irrelevant information, there is also missing information. Corpora of written texts are *a priori* not the same source of input as what children get when they learn their language. Natural language semantics relies on much data that the current VSMs do not have, including multimodal data and frequencies of events too commonplace to be mentioned in writing (Erk, 2016, p.18).

This means that the distributional difference between $\overrightarrow{tell}$ and $\overrightarrow{retell}$ (or $\overrightarrow{marry}$ and $\overrightarrow{remarry}$, or both pairs) does not necessarily reflect the full range of the relevant difference, which could perhaps have helped to bring the vector offset calculation closer to the desired outcome. On this view, in the ideal world all word vectors with the "re-" feature would be nearly aligned. Some blame could also be passed to the condensed vectors such as SVD or neural word embeddings, which blend distributional features in a non-transparent way, potentially obscuring the relevant ones.

The current source corpora and VSMs could certainly be improved. But both linguistics and philosophy suggest that there are also issues with the idea of linguistic relations being so regular.

5.3 Semantics is Messy

In theory, according to the distributional hypothesis, we would expect the relatively straightforward "repeated action" paradigm of verbs with and without the prefix "re-" in Table 2 to surface distributionally in the use of adverbs like "again". However, we have no reason to expect this to happen in quantitatively exactly the same way for all the verbs, even in an "ideal" corpus. And variation would lead to irregularities that we observe.

In fact, such variation would make VSMs more like human mental lexicon, not less. A well-known problem in psychology is the asymmetry of similarity judgments, upon which relational similarity and analogical reasoning are based. Logically *a is like b* is equivalent to *b is like a*, but humans do not necessarily agree with both statements to the same degree (Tversky, 1977).

Consider the "re-" prefix examples above. We could expect 100% success by native English speakers on a "complete the verb paradigm" task, because they would be inevitably made aware of the "add re-" rule during its completion. Even so, processing time would vary due to such factors as frequencies and prototypicality. The psychological evidence is piling for certain gradedness in mental representation of morphological rules: people can rate the same structure differently on complexity ("settlement" is reported more affixed that "government"), similarity judgments for semantically transparent and non-transparent bases are continuous, and there are graded priming effects for both orthographic, semantic and phonological similarity between derived words and their roots (Hay and Baayen, 2005).

There are several connectionist proposals to simulate asymmetry through biases, saliency features, or structural alignment (Thomas and Mareschal, 1997, p.758). The irregularities we observe in the VSMs could perhaps even be welcomed as another way to model this phenomenon - although it remains to be seen to what extent the parallel we draw here is appropriate.

As a side note, let us remember that equations such as $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$ should only be interpreted distributionally, although it is tempting to suppose that they reflect something like semantic features. That would be misleading on several accounts. First of all, the 3CosAdd math is commutative, which would be dubious for semantic features[5]. Secondly, it would bring us to the wall that componential analysis in linguistic semantics has hit a long time ago: semantic features defy definitions[6], they only apply to a portion of vocabulary, and they impose binary oppositions that are psycholinguistically unrealistic (Leech, 1981, pp.117-119).

5.4 Analogy Is Not an Inference Rule

Let us now come back to the fact that the "linguistic regularities" are in fact relying on relational similarity (Section 2), and relational similarity is not something binary. That takes us straight to the most fundamental difficulty with analogy as it is known in philosophy and logic. Analogy is undeniably fundamental to human reasoning as an instrument for discovery and understanding the unknown from the known – but it is not, and has never been an inference rule.

Consider the example where Mars is similar to Earth in several ways, and therefore could be supporting life. This analogy does not guarantee the existence of Martians, and it could even be similarly applied to even less suitable planets.

Basically, the problem with analogy is that not all similarities warrant all conclusions, and establishing valid analogies requires much case-by-case consideration. For this and some other reasons, analogy has long been rejected in generative linguistics as a mechanism for language acquisition through discovery, although now it is making a comeback (Itkonen, 2005, p.67-75).

This general difficulty with analogical reasoning – it does work in humans, but selectively, so to say, – is inherited by the so-called proportional analogies of the $a : a' :: b : b'$ kind. A case in point is their use in schools as verbal reasoning tests. In 2005 analogies were removed from SAT, its criticisms including ambiguity, guesswork and puzzle-like nature (Pringle, 2003). It is also telling that SAT analogy problems came with a set of potential answers to choose from, because otherwise students would supply a range of answers with varying degrees of incorrectness.

In case of the "re-" prefix above, once again, we could expect 100% success rate by humans who could see the "add re-" pattern; but semantic BATS questions would yield more variation. Consider the question "*trout* is to *river* as *lion* is to ___". Some would say *den*, thinking of the river as the trout's "home", but some could say *savanna* in the broader habitat terms; *cage* or *zoo* or *safari park* or even *circus* would all be valid to various degrees. BATS accepts several answer options, but it is hardly feasible to list them all for all cases.

Given the above, the question is: if analogical reasoning requires much case-by-case consideration in humans, what should we expect from VSMs with a single linear algebra operation?

[5] $((\overrightarrow{remarry} - \overrightarrow{marry}) + \overrightarrow{write})$ makes some sense, but $((\overrightarrow{write} - \overrightarrow{marry}) + \overrightarrow{remarry})$ does not.

[6] Is the $\overrightarrow{man} - \overrightarrow{woman}$ result certainly "femaleness" – or perhaps "maleness", or some mysterious "malefemale gender change" semantic feature?

6 Implications for Evaluation of VSMs

The analogy task continues to enjoy immense popularity in the NLP community as the standard evaluation task for VSMs. We have already mentioned two problems with the task: the problem of the Google test scores being flattering to the VSMs (Gladkova et al., 2016), and also 3CosAdd disadvantaging them, because the required semantic information may be encoded in more complex ways (Drozd et al., 2016).

What the present work adds to the discussion is the demonstration of how strongly the accuracy on the analogy task depends on the target vector being relatively close to the source in the vector space model – not only for 3CosAdd, but also 3CosMul and LRCos. This is in fact a fundamental problem that is encountered in many other NLP tasks[7].

That problem brings about the following question: what have we been evaluating with 3CosAdd all this time?

The answer seems to be this: analogy task scores indicate to what extent the semantic space of a given VSM was structured in a way that, for each word category, favored the linguistic relation that happened to be picked by the creators of the particular test dataset. BATS makes this clearer, because it is well balanced across different types of relations. Most models score well on morphological inflections – because morphological forms of the same word are highly distributionally similar and are likely to be close. But we do not see equal success for synonyms, suffixes, colors and other categories – because it is hard to expect of any one model to "guess" which words should have synonyms as closest neighbors and which words should be close to their antonyms.

As a matter of fact, for a general-purpose VSM we would not want that: every word can participate in hundreds of linguistic relations that we may be interested in, but we cannot expect them all to be close neighbors. We would want a VSM whose vector neighborhoods simply reflect whatever distributional properties were observed in a corpus. The challenge is to find reasoning methods that could reliably identify linguistic relations from vectors at any distance.

Given the irregularities discussed in section 5, these methods would also have to rely on a more linguistically and cognitively realistic model of how meanings are reflected in distributional properties of words.

LRCos made a step in the right direction, as it does not rely on unique and neatly aligned word pairs, but it can only work for relations between coherent word classes. That excludes many lexicographic relations like synonyms (*car* is to *automobile* as *snake* is to *serpent*), frame-semantic or encyclopedic relations (*white* is to *snow* as *red* is to *rose*).

7 Conclusion

While it would be highly desirable to have automated reasoning about linguistic relations with VSMs as a powerful, all-purpose tool, it is so far a remote goal. We investigated the potential of the vector offset method in solving the so-called proportional analogies, which rely on one pair of words with a known linguistic relation to identify the missing member of another pair of words.

We have presented a series of experiments showing that the success of the linear vector offset (as well as two better-performing methods) depends on the structure of the VSM: the targets that are further away in the vector space have worse chances of being recovered. This is a crucial limitation: no model could possibly hold all related words close in the vector space, as there are many thousands of linguistic relations, and many are context-dependent.

Furthermore, the offsets of different word vector pairs appear to not be so regular, even for relatively straightforward linguistic relations. We argue that the observed irregularities should not just be blamed on the corpus. There is a number of theoretical issues with the very approach to linguistic relations as something neat and binary. We hope to drive attention to the graded nature of relational similarity that underlies analogical reasoning, and the need for automated reasoning algorithms to become more psychologically plausible in order to become more successful.

Acknowledgements

This work was partially supported by JST CREST Grant number JPMJCR1303, JSPS KAKENHI Grant number JP17K12739, and performed under the auspices of Real-world Big-Data Computation Open Innovation Laboratory, Japan.

[7] E.g. in taxonomy construction it was found helpful to narrow the semantic space with domains or clusters, essentially "zooming in" on certain relations (Fu et al., 2014; Espinosa Anke et al., 2016).

References

Paul Bartha. 2016. Analogy and analogical reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Metaphysics Research Lab, Stanford University. Winter 2016 edition. https://plato.stanford.edu/archives/win2016/entries/reasoning-analogy/.

Trevor Cohen, Dominic Widdows, and Thomas Rindflesch. 2015. Expansion-by-analogy: a vector symbolic approach to semantic search. In *Quantum Interaction*, Springer, pages 54–66. https://doi.org/10.1007/978-3-319-15931-7_5.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. 2015. Splitting compounds by semantic analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*. Charles University in Prague, Praha, Czech Republic, 3-4 September 2015, pages 20–28. http://www.aclweb.org/anthology/W15-5703.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: beyond *king - man + woman = queen*. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 3519–3530. https://www.aclweb.org/anthology/C/C16/C16-1332.pdf.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2012. Cross-language latent relational search between Japanese and English languages using a Web corpus. *ACM Transactions on Asian Language Information Processing (TALIP)* 11(3):11. http://dl.acm.org/citation.cfm?id=2334805.

Katrin Erk. 2016. What do you know about an alligator when you know the company it keeps. *Semantics and Pragmatics* 9(17):1–63. https://doi.org/10.3765/sp.9.17.

Luis Espinosa Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 424–435. https://aclweb.org/anthology/D16-1041.

Stefano Federici, Simonetta Montemagni, and Vito Pirrelli. 1997. Inferring semantic similarity from distributional evidence: An analogy-based approach to word sense disambiguation. In *Proceedings of the ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. pages 90–97. http://aclweb.org/anthology/W/W97/W97-0813.pdf.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 1199–1209. http://202.118.253.69/ rjfu/publications/acl2014.pdf.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW*. ACL, San Diego, California, June 12-17, 2016, pages 47–54. https://doi.org/10.18653/v1/N16-2002.

Jennifer B. Hay and R. Harald Baayen. 2005. Shifting paradigms: Gradient structure in morphology. *Trends in cognitive sciences* 9(7):342–348. https://doi.org/10.1016/j.tics.2005.04.002.

Esa Itkonen. 2005. *Analogy as Structure and Process: Approaches in Linguistic, Cognitive Psychology, and Philosophy of Science*. Number 14 in Human cognitive processing. John Benjamins Pub. Co, Amsterdam ; Philadelphia. https://doi.org/10.1075/hcp.14.

David A. Jurgens, Peter D. Turney, Saif M. Mohammad, and Keith J. Holyoak. 2012. Semeval-2012 task 2: measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics, Montréal, Canada, June 7-8, 2012, pages 356–364. http://dl.acm.org/citation.cfm?id=2387693.

Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde. 2015. Multilingual reliability and "semantic" structure of continuous word spaces. In *Proceedings of the 11th International Conference on Computational Semantics*. Association for Computational Linguistics, pages 40–45. http://www.aclweb.org/anthology/W15-01#page=56.

Siwei Lai, Kang Liu, Liheng Xu, and Jun Zhao. 2016. How to generate a good word embedding? *IEEE Intelligent Systems* 31(6):5–14. https://doi.org/10.1109/MIS.2016.45.

Geoffrey Leech. 1981. *Semantics: The Study of Meaning*. Harmondsworth: Penguin Books.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. pages 171–180. https://doi.org/10.3115/v1/W14-1618.

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. http://www.aclweb.org/anthology/Q15-1016.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the First Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-2503.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Proceedings of International Conference on Learning Representations (ICLR)* http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. pages 3111–3119. http://papers.nips.cc/paper/5021-di.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. 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*. Association for Computational Linguistics, pages 746–751. http://aclweb.org/anthology/N13-1090.

Denis Newman-Griffis, Albert M. Lai, and Eric Fosler-Lussier. 2017. Insights into analogy completion from the biomedical domain. *arXiv:1706.02241 [cs]* http://arxiv.org/abs/1706.02241.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. volume 12, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Paul Pringle. 2003. College board scores with critics of SAT analogies. *Los Angeles Times* http://articles.latimes.com/2003/jul/27/local/me-sat27/2.

Michael SC Thomas and Denis Mareschal. 1997. Connectionism and psychological notions of similarity. In *The Proceedings of the 19th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, Stanford, USA, pages 757–762. http://eprints.bbk.ac.uk/4611/.

Peter Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. pages 482–489. http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=8913366.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics* 32(3):379–416. https://doi.org/10.1162/coli.2006.32.3.379.

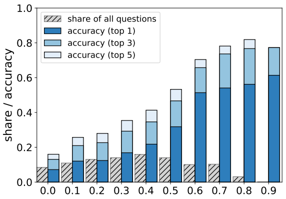
Peter D. Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. pages 905–912. http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=5764174.

Amos Tversky. 1977. Features of similarity. *Psychological Review* 84(4):327–352. https://doi.org/10.1037/0033-295X.84.4.327.

Ekaterina Vylomova, Laura Rimmel, Trevor Cohn, and Timothy Baldwin. 2016. *Take* and *took*, *gaggle* and *goose*, *book* and *read*: evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1671–1682. https://doi.org/10.18653/v1/P16-1158.

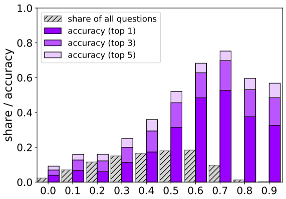
A Supplementary Material

A.1 3CosAdd on GloVe and Word2Vec



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	8.5	7.2	13.1	16.0
0.1 - 0.2	10.9	12.0	21.0	25.7
0.2 - 0.3	13.1	12.4	22.7	28.0
0.3 - 0.4	14.0	16.9	29.2	35.4
0.4 - 0.5	15.9	21.8	34.6	41.3
0.5 - 0.6	14.0	31.8	46.7	53.3
0.6 - 0.7	10.1	51.4	65.7	70.4
0.7 - 0.8	10.3	54.1	73.6	78.2
0.8 - 0.9	3.1	56.2	76.7	81.9
0.9 - 1	0.1	61.4	77.3	77.3

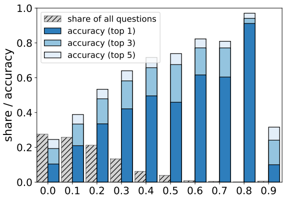
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	2.3	3.9	6.9	9.2
0.1 - 0.2	7.0	6.7	12.7	15.9
0.2 - 0.3	11.5	5.9	12.2	16.0
0.3 - 0.4	15.0	11.4	20.0	25.1
0.4 - 0.5	16.5	17.3	29.4	35.9
0.5 - 0.6	18.0	31.5	45.6	52.1
0.6 - 0.7	18.4	48.4	62.8	68.3
0.7 - 0.8	9.7	52.6	69.7	75.3
0.8 - 0.9	1.3	37.5	53.1	59.7
0.9 - 1	0.3	32.6	48.5	56.8

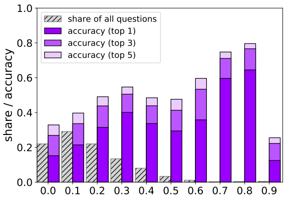
(b) Word2Vec

Figure 9: Similarity between vectors a and a'



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	27.6	10.4	19.3	24.5
0.1 - 0.2	25.8	20.9	33.4	38.8
0.2 - 0.3	21.2	33.5	47.9	53.4
0.3 - 0.4	13.3	42.1	58.2	64.0
0.4 - 0.5	6.2	49.6	65.7	71.6
0.5 - 0.6	3.9	45.9	67.5	73.8
0.6 - 0.7	0.9	61.6	77.2	82.3
0.7 - 0.8	0.5	60.4	77.1	80.9
0.8 - 0.9	0.0	91.2	94.1	97.1
0.9 - 1	0.6	10.0	24.1	31.6

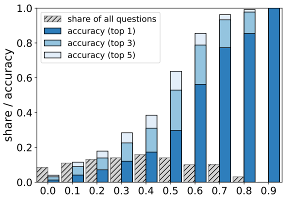
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	22.0	15.2	26.9	32.9
0.1 - 0.2	29.1	21.4	33.6	39.7
0.2 - 0.3	22.0	31.5	43.8	49.1
0.3 - 0.4	13.4	40.1	50.5	54.7
0.4 - 0.5	8.0	33.7	43.9	48.5
0.5 - 0.6	3.4	29.4	41.3	47.6
0.6 - 0.7	1.1	35.8	53.4	59.6
0.7 - 0.8	0.3	59.6	71.1	74.8
0.8 - 0.9	0.4	64.5	76.6	79.6
0.9 - 1	0.4	12.5	22.3	25.5

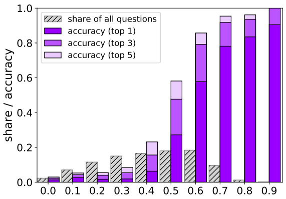
(b) Word2Vec

Figure 10: Similarity between vectors a' and b'



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	8.5	1.3	3.1	4.0
0.1 - 0.2	11.0	4.1	9.0	11.5
0.2 - 0.3	13.1	7.1	13.9	17.9
0.3 - 0.4	14.0	12.1	22.6	28.4
0.4 - 0.5	15.9	17.3	31.0	38.5
0.5 - 0.6	14.0	29.7	52.9	63.7
0.6 - 0.7	10.1	56.2	78.8	85.5
0.7 - 0.8	10.3	77.3	93.3	96.3
0.8 - 0.9	3.1	85.5	97.8	99.2
0.9 - 1	0.0	–	–	–

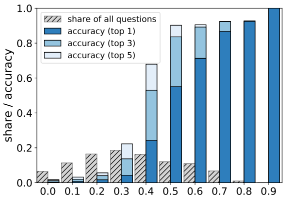
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	2.3	1.3	2.3	3.0
0.1 - 0.2	7.0	2.7	4.3	5.3
0.2 - 0.3	11.6	1.7	4.0	5.5
0.3 - 0.4	15.0	1.9	5.5	8.4
0.4 - 0.5	16.5	6.3	15.6	23.2
0.5 - 0.6	18.0	27.2	47.7	58.1
0.6 - 0.7	18.4	57.8	79.2	85.7
0.7 - 0.8	9.7	78.1	91.8	95.4
0.8 - 0.9	1.3	83.5	93.6	96.2
0.9 - 1	0.2	90.5	100	100

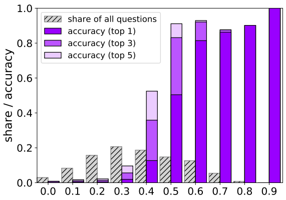
(b) Word2Vec

Figure 11: Similarity between vectors b and b'



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	6.6	0.4	1.2	1.7
0.1 - 0.2	11.3	0.7	2.0	3.2
0.2 - 0.3	16.4	1.7	4.0	5.7
0.3 - 0.4	18.6	4.2	13.7	22.3
0.4 - 0.5	16.2	24.3	53.0	67.9
0.5 - 0.6	12.0	55.0	83.6	90.2
0.6 - 0.7	11.0	71.3	89.2	90.5
0.7 - 0.8	6.8	86.7	92.3	92.5
0.8 - 0.9	1.0	92.5	92.7	92.8
0.9 - 1	0.1	100	100	100

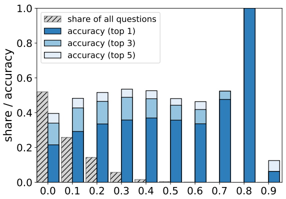
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	3.0	0.1	0.7	0.9
0.1 - 0.2	8.3	0.4	1.1	1.7
0.2 - 0.3	15.7	0.4	1.3	2.2
0.3 - 0.4	20.7	1.9	5.6	9.6
0.4 - 0.5	18.7	12.7	35.9	52.5
0.5 - 0.6	14.8	50.4	83.2	91.1
0.6 - 0.7	12.6	81.4	92.1	93.0
0.7 - 0.8	5.5	86.3	87.6	87.7
0.8 - 0.9	0.7	90.2	90.2	90.2
0.9 - 1	0.0	100	100	100

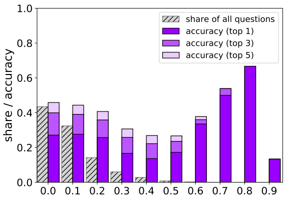
(b) Word2Vec

Figure 12: Similarity between vector b' and predicted vector



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	52.0	21.5	34.0	39.6
0.1 - 0.2	25.8	29.2	42.7	48.3
0.2 - 0.3	14.3	33.5	46.5	51.6
0.3 - 0.4	5.8	35.8	48.9	53.5
0.4 - 0.5	1.6	37.0	48.0	52.6
0.5 - 0.6	0.4	35.7	44.3	48.1
0.6 - 0.7	0.1	33.6	41.8	46.4
0.7 - 0.8	0.0	47.6	52.4	52.4
0.8 - 0.9	0.0	–	–	–
0.9 - 1	0.0	6.2	6.2	12.5

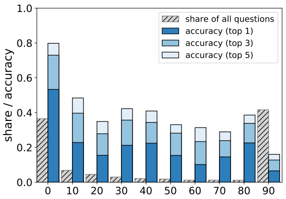
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	43.5	27.1	39.9	45.8
0.1 - 0.2	32.4	27.6	39.1	44.3
0.2 - 0.3	14.2	25.7	35.8	40.7
0.3 - 0.4	6.0	16.7	25.8	30.7
0.4 - 0.5	2.8	13.6	22.2	26.9
0.5 - 0.6	0.8	17.2	23.5	26.7
0.6 - 0.7	0.2	33.5	36.0	37.8
0.7 - 0.8	0.0	50.0	53.8	53.8
0.8 - 0.9	0.0	66.7	66.7	66.7
0.9 - 1	0.0	13.3	13.3	13.3

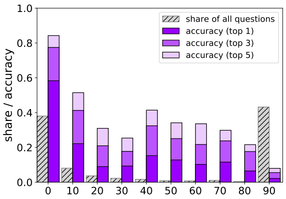
(b) Word2Vec

Figure 13: Similarity between vector b' and a



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 10	36.5	53.3	73.0	79.8
10 - 20	6.8	22.8	39.7	48.4
20 - 30	4.5	15.5	27.8	34.9
30 - 40	3.0	21.2	35.7	42.3
40 - 50	2.1	22.4	34.4	40.9
50 - 60	1.8	15.4	28.2	33.0
60 - 70	1.2	10.2	23.4	31.4
70 - 80	1.2	14.6	23.9	28.9
80 - 90	1.2	22.6	33.8	38.6
90 - 100	41.6	6.5	12.7	16.0

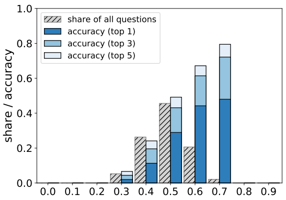
(a) GloVe



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 10	38.0	58.3	77.5	84.3
10 - 20	8.1	22.1	41.3	51.4
20 - 30	3.7	9.0	20.9	31.0
30 - 40	2.3	9.3	17.8	25.4
40 - 50	1.7	15.3	32.4	41.4
50 - 60	0.9	12.8	25.1	34.1
60 - 70	0.8	10.2	21.7	33.6
70 - 80	1.1	11.6	23.8	29.8
80 - 90	0.3	6.5	17.6	21.6
90 - 100	43.2	2.2	5.5	8.0

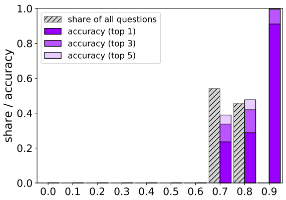
(b) Word2Vec

Figure 14: The rank of b in the neighborhood of b'



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	0.0	–	–	–
0.1 - 0.2	0.0	–	–	–
0.2 - 0.3	0.0	0.0	0.0	0.0
0.3 - 0.4	5.3	2.0	4.4	6.6
0.4 - 0.5	26.3	11.3	19.6	24.1
0.5 - 0.6	45.6	28.9	43.1	49.2
0.6 - 0.7	20.6	44.2	61.4	67.2
0.7 - 0.8	2.0	48.0	72.1	79.5
0.8 - 0.9	0.0	0.0	0.0	0.0
0.9 - 1	0.0	0.0	0.0	0.0

(a) GloVe

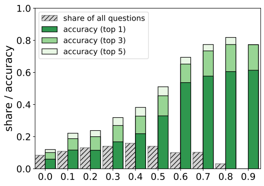


Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	0.0	–	–	–
0.1 - 0.2	0.0	–	–	–
0.2 - 0.3	0.0	–	–	–
0.3 - 0.4	0.0	–	–	–
0.4 - 0.5	0.0	–	–	–
0.5 - 0.6	0.0	–	–	–
0.6 - 0.7	0.1	0.0	0.0	0.0
0.7 - 0.8	54.1	23.5	33.8	38.9
0.8 - 0.9	45.7	28.7	41.9	47.6
0.9 - 1	0.2	91.1	99.3	100

(b) Word2Vec

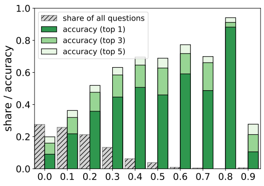
Figure 15: Similarity between b' and its 5th neighbor

A.2 3CosMul on GloVe



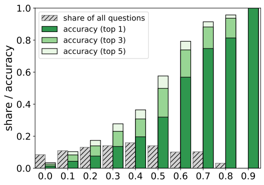
Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	8.5	6.0	10.1	12.0
0.1 - 0.2	10.9	11.7	18.8	22.1
0.2 - 0.3	13.1	11.5	20.0	23.8
0.3 - 0.4	14.0	16.8	26.9	31.9
0.4 - 0.5	15.9	21.9	32.8	38.2
0.5 - 0.6	14.0	33.0	45.5	51.1
0.6 - 0.7	10.1	53.7	65.2	69.4
0.7 - 0.8	10.3	57.7	73.4	77.4
0.8 - 0.9	3.1	60.5	77.4	81.8
0.9 - 1	0.1	61.4	77.3	77.3

Figure 16: Similarity between vectors a and a'



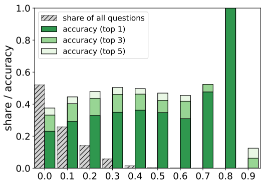
Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	27.6	9.1	16.0	19.9
0.1 - 0.2	25.8	21.8	31.9	36.3
0.2 - 0.3	21.2	35.8	47.6	51.9
0.3 - 0.4	13.3	44.7	58.4	63.1
0.4 - 0.5	6.2	50.7	64.6	69.6
0.5 - 0.6	3.9	46.0	62.7	68.9
0.6 - 0.7	0.9	59.1	71.8	77.3
0.7 - 0.8	0.5	48.7	66.1	69.9
0.8 - 0.9	0.0	88.2	91.2	94.1
0.9 - 1	0.6	10.6	21.3	27.8

Figure 17: Similarity between vectors a' and b'



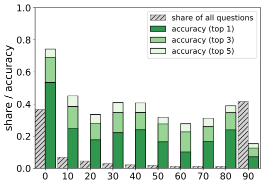
Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	8.5	1.4	2.6	3.5
0.1 - 0.2	11.0	4.3	8.3	10.3
0.2 - 0.3	13.1	7.6	13.9	17.4
0.3 - 0.4	14.0	13.6	23.0	27.7
0.4 - 0.5	15.9	19.6	30.8	36.4
0.5 - 0.6	14.0	31.9	49.9	57.6
0.6 - 0.7	10.1	56.9	73.9	79.3
0.7 - 0.8	10.3	74.8	88.2	91.4
0.8 - 0.9	3.1	81.3	93.7	95.7
0.9 - 1	0.0	–	–	–

Figure 18: Similarity between vectors b and b'



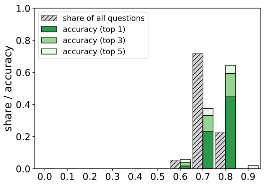
Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	52.0	23.1	33.2	37.6
0.1 - 0.2	25.8	29.2	40.3	44.5
0.2 - 0.3	14.3	32.9	43.7	48.0
0.3 - 0.4	5.8	34.9	46.2	50.4
0.4 - 0.5	1.6	36.2	46.3	49.7
0.5 - 0.6	0.4	34.8	42.4	46.9
0.6 - 0.7	0.1	30.9	41.8	45.5
0.7 - 0.8	0.0	47.6	52.4	52.4
0.8 - 0.9	0.0	–	–	–
0.9 - 1	0.0	0.0	6.2	12.5

Figure 19: Similarity between vector b' and a



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 10	36.5	53.5	68.9	74.3
10 - 20	6.8	25.0	38.5	45.1
20 - 30	4.5	17.7	28.1	33.5
30 - 40	3.0	22.2	34.8	40.9
40 - 50	2.1	24.0	34.7	40.7
50 - 60	1.8	16.5	27.6	31.8
60 - 70	1.2	10.2	22.7	27.8
70 - 80	1.2	16.9	25.9	31.2
80 - 90	1.2	23.9	34.6	38.9
90 - 100	41.7	7.2	12.6	15.3

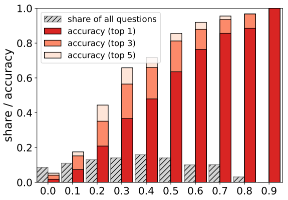
Figure 20: The rank of b in the neighborhood of b'



Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	0.0	–	–	–
0.1 - 0.2	0.0	–	–	–
0.2 - 0.3	0.0	–	–	–
0.3 - 0.4	0.0	–	–	–
0.4 - 0.5	0.0	–	–	–
0.5 - 0.6	0.0	–	–	–
0.6 - 0.7	5.3	1.7	3.9	5.7
0.7 - 0.8	71.9	23.3	33.1	37.4
0.8 - 0.9	22.7	44.8	59.4	64.5
0.9 - 1	0.1	0.0	0.0	2.1

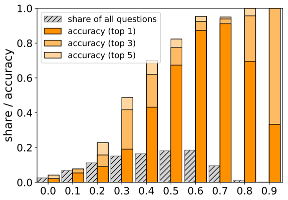
Figure 21: Similarity between b' and its 5th neighbor

A.3 LRCos on GloVe and Word2Vec



(a) GloVe

Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0 - 0.1	8.6	1.8	4.1	5.3
0.1 - 0.2	11.0	7.4	15.2	17.5
0.2 - 0.3	13.1	20.8	35.1	44.4
0.3 - 0.4	14.1	36.7	56.5	65.8
0.4 - 0.5	15.9	47.9	66.0	71.7
0.5 - 0.6	14.0	63.5	81.2	85.6
0.6 - 0.7	10.1	76.4	87.9	92.0
0.7 - 0.8	10.2	85.6	93.6	95.5
0.8 - 0.9	3.1	88.5	96.7	96.7
0.9 - 1	0.0	–	–	–

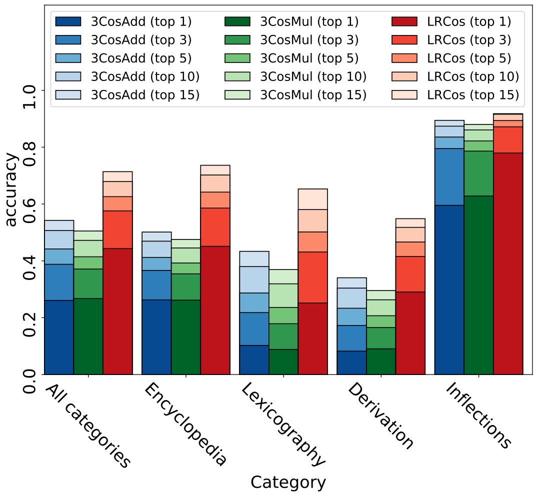


(b) Word2Vec

Similarity Bin	Share	Accuracy (%)		
		top 1	top 3	top 5
0.0 - 0.1	2.6	2.1	2.1	4.2
0.1 - 0.2	6.9	5.4	7.7	7.7
0.2 - 0.3	11.2	9.0	15.7	22.9
0.3 - 0.4	15.1	19.1	41.7	48.8
0.4 - 0.5	16.5	43.2	62.0	70.1
0.5 - 0.6	18.2	67.4	77.4	82.4
0.6 - 0.7	18.5	87.3	92.5	95.4
0.7 - 0.8	9.6	91.1	93.9	95.0
0.8 - 0.9	1.2	69.6	95.7	100
0.9 - 1	0.2	33.3	100	100

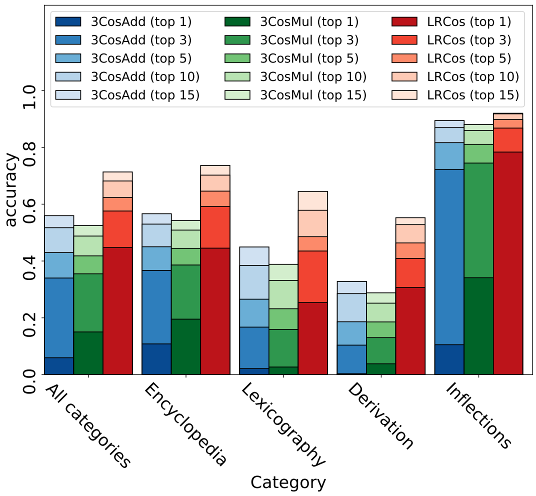
Figure 22: Similarity between vectors b and b'

A.4 Comparison between 3CosAdd, 3CosMul and LRCos on GloVe



Category	Method	Accuracy (%)				
		top 1	top 3	top 5	top 10	top 15
All	3CosAdd	26.1	38.8	44.2	50.7	54.2
	3CosMul	26.7	37.1	41.4	47.2	50.5
	LRCos	44.3	57.6	62.6	67.9	71.4
Encyclopedia	3CosAdd	26.3	36.6	41.2	46.9	50.1
	3CosMul	26.2	35.4	39.2	44.5	47.5
	LRCos	45.1	58.6	64.2	70.2	73.6
Lexicography	3CosAdd	10.2	21.8	28.7	38.0	43.3
	3CosMul	8.8	17.9	23.6	31.9	36.9
	LRCos	25.2	43.1	50.2	58.0	65.3
Derivation	3CosAdd	8.2	17.3	23.3	30.4	34.1
	3CosMul	9.1	16.5	20.7	26.3	29.5
	LRCos	29.0	41.5	46.6	51.8	54.8
Inflections	3CosAdd	59.5	79.5	83.6	87.4	89.4
	3CosMul	62.8	78.6	82.2	86.1	88.0
	LRCos	77.9	87.2	89.4	91.6	91.8

Figure 23: 3CosAdd vs 3CosMul vs LRCos



Category	Method	Accuracy (%)				
		top 1	top 3	top 5	top 10	top 15
All	3CosAdd	6.0	34.0	43.0	51.7	56.0
	3CosMul	15.0	35.5	41.8	48.8	52.5
	LRCos	44.7	57.6	62.3	68.2	71.3
Encyclopedia	3CosAdd	10.8	36.7	45.0	53.0	56.6
	3CosMul	19.5	38.6	44.4	50.9	54.2
	LRCos	44.5	59.2	64.6	70.2	73.6
Lexicography	3CosAdd	2.1	16.7	26.6	38.4	44.9
	3CosMul	2.7	15.9	23.2	33.2	38.8
	LRCos	25.4	43.5	48.6	57.8	64.5
Derivation	3CosAdd	0.3	10.4	18.6	28.5	32.8
	3CosMul	3.8	13.0	18.6	25.2	28.8
	LRCos	30.7	40.9	46.4	52.8	55.2
Inflections	3CosAdd	10.5	72.2	81.6	86.9	89.4
	3CosMul	34.1	74.5	81.0	85.9	88.0
	LRCos	78.3	86.8	89.8	91.8	92.0

Figure 24: 3CosAdd vs 3CosMul vs LRCos (“honest” version)