Fitting Semantic Relations to Word Embeddings

Eric Kafe
MegaDoc
Charlottenlund, Denmark
kafe@megadoc.net

Abstract

We fit WordNet relations to word embeddings, using *3CosAvg* and *LRCos*, two set-based methods for analogy resolution, and introduce *3CosWeight*, a new, *weighted* variant of *3CosAvg*. We test the performance of the resulting semantic vectors in *lexicographic semantics tests*, and show that none of the tested classifiers can learn symmetric relations like *synonymy* and *antonymy*, since the source and target words of these relations are the same set. By contrast, with the asymmetric relations (*hyperonymy / hyponymy* and *meronymy*), both *3CosAvg* and *LRCos* clearly outperform the baseline in all cases, while *3CosWeight* attained the best scores with *hyponymy* and *meronymy*, suggesting that this new method could provide a useful alternative to previous approaches.

1 Introduction

Analogy is the prototypical formulation of any relation: *a is to a' as b is to b'* means that the relation between a and a' is the same as the relation between b and b'. Thus, the analogy establishes a paradigmatic relation between a class of source items (a and b) and a class of target items (a' and b'), and all relations are special cases of analogy.

Both morphological analogies like $(car, cars) \approx (apple, apples)$, and semantic analogies like $(man, woman) \approx (king, queen)$ have been shown to hold in vector-space representations of words, derived from cooccurrence matrices in large corpora (Mikolov et al., 2013c). This approach has proven useful in many applications, in particular machine translation, where it reveals analogies across languages (Mikolov et al., 2013a), although more complex morphology or deeper semantic relations cause a drop in accuracy (Köper et al., 2015).

The original method (Mikolov et al., 2013c), which is now called *3CosAdd*, resolved *analogy completion* tasks like $(man, king) \approx (woman, ?)$ by searching for the most similar vector to $woman + king - man$, using *cosine similarity*, with $queen$, as result.

3CosAdd:

$$b' = argmax_{b' \in V}(cos(b', b + a' - a)) \qquad (1)$$

Alternative methods like *PairDistance* and *3CosMul* have been shown to occasionally perform slightly better (Levy et al., 2015).

Very often, the most similar target word b' is likely to be one of the already given words a, a' and especially b, so these are always discarded from the searched vocabulary V, which should, more precisely, be understood as $C^V_{a,a',b}$ (the complement set of the three premisses in the vocabulary). Otherwise, test accuracy often drops to *zero* (Linzen, 2016), raising questions about the proper interpretation of these vector-space operations (Rogers et al., 2017; Schluter, 2018).

However, the limits of *pair-based* approaches became clear with the *Bigger Analogy Test Sets (BATS)* (Gladkova et al., 2016), where, in particular, a series of *Lexicographic semantics* tests proved very difficult. These tests consist in ten series of questions, covering seven semantic relations (hypernyms, hyponyms, three kinds of meronyms, synonyms and antonyms). The first example from each series is shown in Tab. 1, where we can see that the expected answer often differs from the corresponding WordNet target. In particular, four out of these ten examples do not have a solution in WordNet 3.1, which adds to the difficulty of solving these tests.

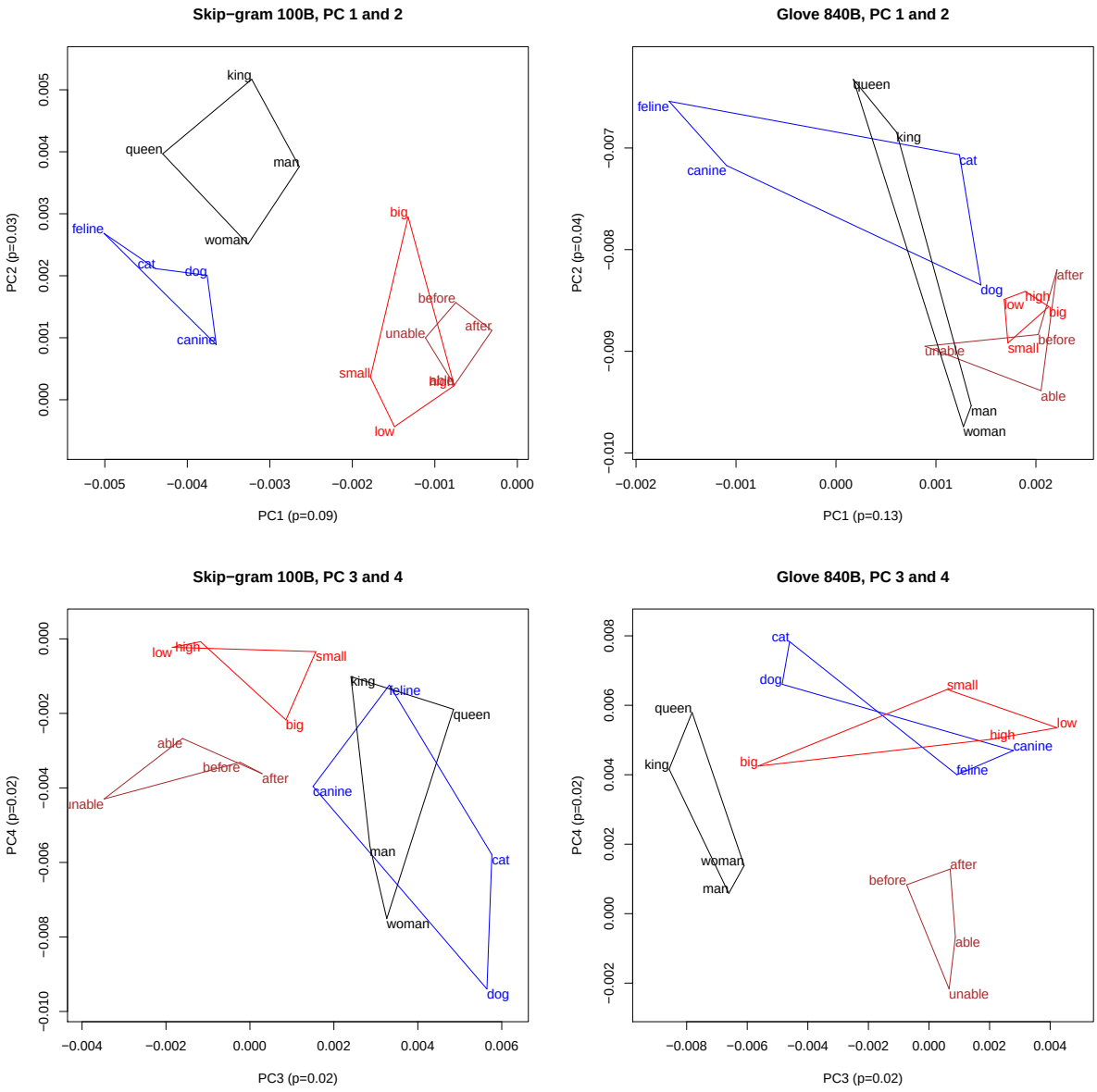

Figure 1: Word Analogies in Skip-gram and Glove models (Principal Components)

A new standard was introduced with the *set-based* methods *3CosAvg* and *LRCos* (Drozd et al., 2016). Instead of relying only on two pairs of words, these methods solve analogies by learning from several pairs, which was shown to clearly outperform all previous methods, although the performance on the *Lexicographic semantics* tests remained modest.

On the other hand, using the semantic knowledge from WordNet relations as a training objective of word embeddings has been shown to improve their performance on semantic tasks (Yu and Dredze, 2014), and the hypernymy and meronymy relations of the Polish wordnet have been successfully used to train linear classifiers (Czachor et al., 2018). A complementary approach consists in retrofitting the embeddings to the semantic relations, which improved on previous baselines (Faruqui et al., 2014), although it seems unlikely that retrofitting can benefit other words than those that were retrofitted.

In this study, we apply the set-based approach to the WordNet relations (Fellbaum, 1998), by using *3CosAvg* and *LRCos* to fit WordNet relations to word embeddings, and test the performance of the resulting vectors on the *Lexicographic semantics tests* from BATS.

Table 1: Lexicographic test examples from BATS

TEST	QUESTION	ACCEPTED ANSWERS	WORDNET 3.1	
L01 [HYPERNYMS - ANIMALS]	*allosaurus*	*dinosaur, reptile, bird, archosaur, archosaurian, archosaurian reptile,*	HYPERNYM	bird-footed dinosaur, theropod, theropod dinosaur
L02 [HYPERNYMS - MISC]	*armchair*	*chair, seat, piece of furniture, article of furniture, furnishing, artifact, artefact, unit, object, physical object, physical entity, entity*	HYPERNYM	chair
L03 [HYPONYMS - MISC]	*backpack*	*daypack, kitbag, kit bag*	HYPONYM	kit bag, kitbag
L04 [MERONYMS - SUBSTANCE]	*atmosphere*	*gas, oxygen, hydrogen, nitrogen, ozone*	HAS SUBSTANCE	∅
L05 [MERONYMS - MEMBER]	*acrobat*	*troupe*	IS MEMBER	∅
L06 [MERONYMS - PART]	*academia*	*college, university, institute*	HAS PART	college, university
L07 [SYNONYMS - INTENSITY]	*afraid*	*terrified, horrified, scared, stiff, petrified, fearful, panicky*	SYNONYM	∅
L08 [SYNONYMS - EXACT]	*airplane*	*aeroplane, plane*	SYNONYM	aeroplane, plane
L09 [ANTONYMS - GRADABLE]	*able*	*unable, incapable, incompetent, unequal*	ANTONYM	unable
L10 [ANTONYMS - BINARY]	*after*	*before, earlier, previously*	ANTONYM	∅

2 Methods

2.1 Set-based analogy resolution

We test the set-based methods *3CosAvg* and *LRCos* (Drozd et al., 2016), and compare their performance with the *Only-B* baseline (Linzen, 2016), and with a new, weighted formulation of *3CosAvg*, which we call *3CosWeight*.

Only-B (Linzen, 2016) is a very appropriate baseline, because it simply disregards the training set, so it allows to precisely gauge the advantage obtained from set-based approaches:

$$b' = argmax_{b' \in V}(cos(b', b)) \quad (2)$$

As always, words that are already known (here only b) need to be discarded from the searched vocabulary V.

Add-Opposite (Linzen, 2016) tests the opposite direction of *3CosAdd* (Eq. 1):

$$b' = argmax_{b' \in V}(cos(b', b + a - a')) \quad (3)$$

3CosAvg (Drozd et al., 2016) is an extension of *3CosAdd*, which, instead of a single word pair, uses the difference between the overall average of the source and target classes:

$$b' = argmax_{b' \in V}(cos(b', b + avg\_offset)) \quad (4)$$

$$avg\_offset^1 = \frac{\sum_{i=0}^{m} a'_i}{m} - \frac{\sum_{i=0}^{n} a_i}{n} \quad (5)$$

A slightly different variation of *3CosAvg* calculates *avg_offset* as the average of the vector differences in each (source,target) pair instead of the difference between the overall class averages (Bouraoui et al., 2018). Thus, the practical implementation of *3CosAvg* is open to various interpretations and extensions, as we will see next.

3CosWeight is a new, weighted formulation of *3CosAvg*, where we multiply the previously defined *avg_offset* with a weight *w*:

$$b' = argmax_{b' \in V}(cos(b', b + (w * avg\_offset))) \quad (6)$$

[1]Thanks to Aleksander Drozd, who gave us permission to correct the order of the subtraction in the *avg_offset* formula (Eq. 5). The formula printed in the original article (Drozd et al., 2016) unfortunately presents this subtraction in the opposite order (a *minus* a').

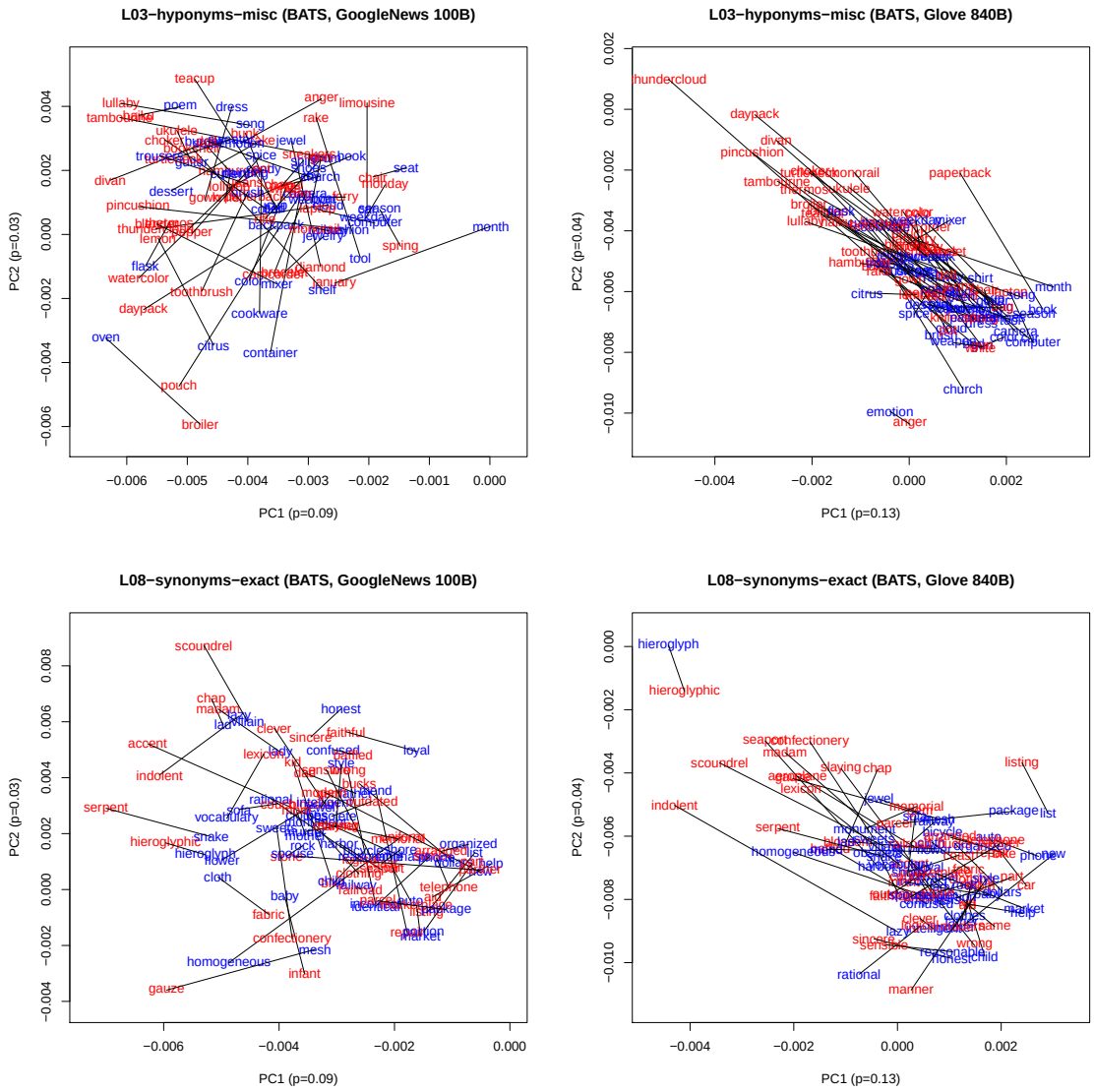

Figure 2: BATS relation pairs in GoogleNews and Glove (Principal Components)

It follows from this definition that *3CosWeight* is identical to *3CosAvg* when multiplying the averaged vector by $w = 1$, and that the result is identical to the *Only-B* method when $w = 0$, while multiplying by $w = -1$ is identical to adding the opposite vector, like in the *Add-Opposite* method. In this study, we try whole integer values of *w* in the range [-2,+5], in order to test whether the weight *w* can boost the performance of the averaged vectors.

Last, we compare these results with *LRCos* (Drozd et al., 2016).

LRCos uses logistic regression to calculate the probability that b' belongs to the target class:

$$b' = argmax_{b' \in V}(P_{(b' \in target\_class)} * cos(b', b)) \quad (7)$$

2.2 Implementation

We downloaded two widely-known sets of embeddings, which have emerged as the best performers in various benchmarks, and are freely available online. Both rely on very large corpora and consist in word vectors with 300 dimensions, meaning that each vector is an array of 300 floating-point numbers in the interval [-1, +1].

The *GoogleNews-vectors-negative300* embeddings [2] are *Skip-gram vectors* (Mikolov et al., 2013b), representing a corpus of 100 billion words, while the *glove.840B.300d* [3] embeddings

[2] https://s3.amazonaws.com/dl4j-distribution/GoogleNews-vectors-negative300.bin.gz

[3] http://nlp.stanford.edu/data/glove.840B.300d.zip

consist in *Global Vectors* (Pennington et al., 2014), derived from a corpus of 840 billion words.

For each of the *Lexicographic semantics* relations in BATS, we produced a two-column text database with the word pairs from the corresponding WordNet 3.1 relation converted to lowercase.

We used the open-source *Vecto v. 0.2* [4] software package to load and process the embeddings, and perform the BATS tests. First, we applied Vecto's *filter_by_vocab* function in order to restrict the embeddings to the set union of all WordNet relations (147478 words) and the words in the BATS *Lexicographic semantics* tests (4126 words), converted to lowercase, yielding a vocabulary of 147620 words, of which 54697 were present in the GoogleNews embeddings, and 65066 in Glove. Thus, although both of the original embeddings include over two million "words" (many of which are noise), they actually cover less than half of the WordNet vocabulary.

We wrote a small Python dictionary called *bats2wn*, which links the adequate WordNet relation (hypernyms, hyponyms, meronyms, synonyms or antonyms) to each of the *Lexicographic semantics* tests in BATS (cf. Tab. 1), so that this data can be processed by the analogy resolution methods, where we simply replace the BATS training set by the corresponding WordNet relation pairs. This only required very small additions to the original Python code in *Vecto*.

Contrary to the BATS pairs, where each target is a list, in our WordNet relation pairs, each target is only a single word. So although the current version *v. 0.2* of Vecto uses a heuristic to speed up learning by only considering the first valid word in each target list, this short-cut has no effect here, because each relation pair only contains one local target, so all targets of each source word are actually used. This allows to preserve the symmetry of the symmetrical relations (synonymy and antonymy), which would otherwise be compromised by the arbitrary loss of some targets.

We became aware of this potential problem by first using WordNet relation pairs converted to the BATS target list format, and realizing that the results did not have the expected properties: hypernymy and hyponymy could not be recognized as inverse relations, and synonymy and antonymy were not symmetric. So this problem was solved by presenting the relation data as word pairs instead of target lists, without modifying Vecto, which would require removing a *break* statement in the *3CosAvg* implementation, and merging the target lists for *LRCos*.

It is important to note that the current (*v. 0.2*) Vecto implementation of *avg_offset* differs from the article formula (Drozd et al., 2016) by also averaging over the *m* local targets of each source word, before calculating the global difference of averages (Eq. 8). More precisely formulated, the global target class average is thus the average of the local averages.

$$avg\_offset = \frac{\sum_{i=0}^{n} \frac{\sum_{j=0}^{m} a'_j}{m} - a_i}{n} \qquad (8)$$

Normally, this detail would result in small variations, compared to implementations that only subtract the global averages. However, the current Vecto implementation only picks one word in the target list, so the local averaging has no effect, since it only averages over a single word. In our setup, each relation pair is also presented with only one target, but all target words are used, so the result is actually equivalent to the original formula (Eq. 5), and the mathematical properties of the studied relations are preserved.

With some tests, the *LRCos* precision could vary by a few percent between subsequent runs, because Vecto's standard implementation relies on random words for the negative examples used for training the classifier. Specifically, Vecto (version 0.2) uses the target word of each relation pair as *positive* examples, while the *negative* examples consist in four copies of the source words of the relation, plus a set of random words of the same size as the set of source words. Since the arbitrary random choices can be fortunate for one embedding and unlucky for another, the standard implementation of LRCos does not allow fair comparisons. So we also tested a deterministic variant of LRCos, where we simply removed the random part of the *negative* examples.

We used the default settings in Vecto to perform series of *Leave-one-out* cross-validations, where each question is answered after training on all the (source, target) pairs in the tested semantic relation, where the question word is not a source word.

[4] https://github.com/vecto-ai

Table 2: WordNet relations fitted with *3CosAvg* to 300-dim. Skip-gram and Glove vectors

	SKIP-GRAM					GLOVE				
dim.	1	2	...	299	300	1	2	...	299	300
HYPERNYM	-0.001645	0.000994	...	-0.000552	-0.009935	-0.011362	0.011182	...	0.003938	0.006483
HYPONYM	0.001644	-0.000996	...	0.000557	0.009932	0.011361	-0.011181	...	-0.003940	-0.006482
HASSUBSTANCE	-0.007939	0.002562	...	0.003922	0.004445	-0.012542	0.015891	...	0.001864	0.018110
ISMEMBER	-0.005050	-0.002306	...	0.018941	-0.006816	-0.003275	-0.000562	...	-0.000785	-0.003439
HASPART	-0.005742	-0.001865	...	-0.004137	0.003926	0.015685	-0.003739	...	0.005990	-0.015377
SYNONYM	-0.000001	-0.000001	...	-0.000001	-0.000004	0.000002	0.000009	...	0.000002	-0.000001
ANTONYM	-0.000008	0.000009	...	-0.000021	0.000016	-0.000008	-0.000015	...	-0.000029	-0.000008

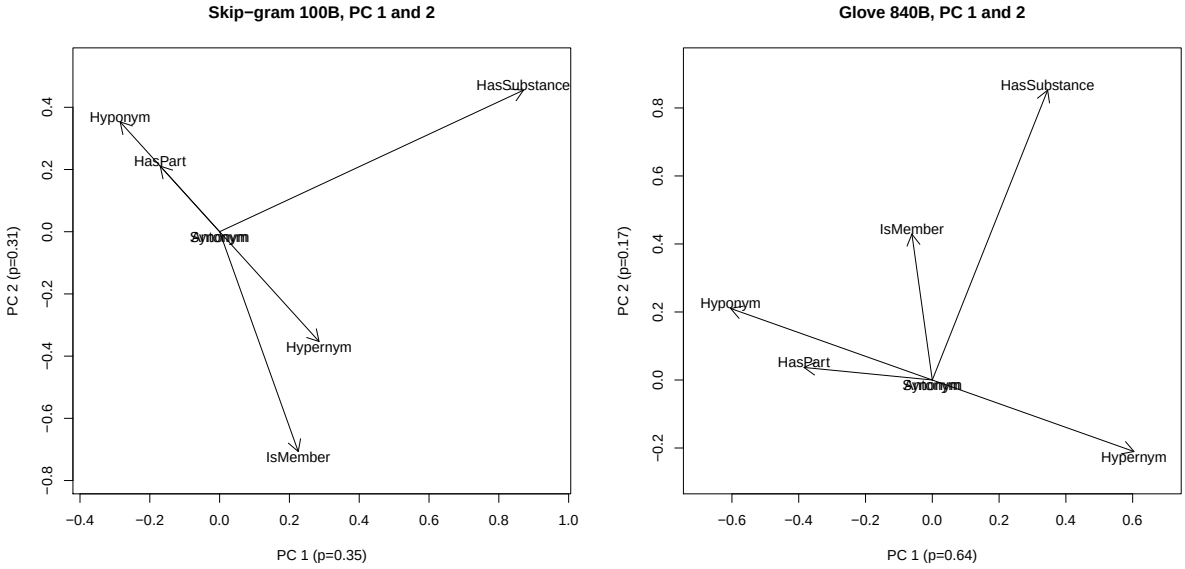

Figure 3: Fitted WordNet relation vectors (Principal Components of Tab. 2)

3 Results

3.1 WordNet vectors

Applying *3CosAvg* (Eq. 4) on WordNet relation pairs in the Skip-gram and Glove embeddings produced the set of semantic *WordNet vectors* shown in Tab. 2.

Like the word vectors, each WordNet relation vector is a list of 300 real numbers in the interval [-1,+1], representing the average projection from the relation source words to their related targets. In both cases we see that each value in the respective vectors of the inverse relations (hypernymy and hyponymy) are the negative of each other up to the fifth decimal, while the sixth decimal shows a spurious divergence, due to the inherent inaccuracy of floating-point arithmetics. As mentioned earlier, this important property of the inverse relations may be lost when using heuristics to prune the training set, which we avoided here by presenting the relations as word pairs instead of target lists.

In theory, the vectors for the symmetric relations (synonymy and antonymy) should contain only zeroes, since the set of source words is identical to the set of target words, so the difference of their respective set averages is expected to be exactly *zero*. In practice, the *synonymy vectors* contain only zeroes up to the fifth decimal (cf. Tab. 2), while the sixth decimals reveal errors introduced by floating-point operations. By contrast, with arbitrary target list pruning, the non-zero values would already appear at the third decimal. Exceptionally, the *antonymy vectors* contain a few non-zero values at the fifth decimal, thus revealing a small error in WordNet 3.1, where a few antonym pairs (for ex. *have* vs. *lack* and *lack* vs. *miss*) do not have a symmetric variant.

3.1.1 Visualizing the vectors

We performed a Principal Components Analysis of the subset of the Glove and GoogleNews em-

Table 3: Precision with WN 3.1 vectors (percent)

	SKIP-GRAM										GLOVE									
	3CosWeight								*LRCos*		*3CosWeight*								*LRCos*	
weight	-2	-1	0	1	2	3	4	5	det	rnd	-2	-1	0	1	2	3	4	5	det	rnd
L01: *Hypernym*	6	8	10	16	20	26	30	36	50	46	4	6	8	10	20	34	36	30	**66**	56
L02: *Hypernym*	0	0	2	4	4	6	6	6	20	14	2	6	10	12	16	16	22	24	**36**	**36**
L03: *Hyponym*	18	22	28	30	30	26	30	32	20	22	22	22	24	32	38	**42**	38	30	32	30
L04: *Substance*	0	0	0	2	2	2	0	0	4	6	6	8	8	10	10	8	8	**12**	2	**12**
L05: *Member*	4	4	4	6	6	10	10	10	6	12	2	6	8	8	12	12	**14**	12	8	10
L06: *Parts*	6	6	6	6	6	6	6	6	12	14	2	2	2	6	8	10	14	**18**	16	16
L07: *Synonym*	26	26	26	26	26	26	26	26	26	**28**	22	22	22	22	22	22	22	22	22	24
L08: *Synonym*	28	28	28	28	28	28	28	28	28	36	44	44	44	44	44	44	44	44	44	**46**
L09: *Antonym*	18	18	18	18	18	18	18	18	18	**22**	14	14	14	14	14	14	14	14	14	14
L10: *Antonym*	38	38	38	38	38	38	38	38	32	30	**48**	**48**	**48**	**48**	**48**	**48**	**48**	**48**	**48**	36
mean	14.4	15	16	17.4	17.8	18.6	19.2	20	21.6	23	16.6	17.8	18.8	20.6	23.2	25	26	25.4	**28.8**	28

beddings used in this study, i. e. the union set of the WordNet and BATS vocabularies. Fig. 1 shows some well-known **word analogies** plotted onto their principal components. The proportion of variance explained by each component is indicated in parentheses, and we see that it is low. For example, with the two first components (PC1 and PC2) of the Skip-gram model, the cumulated proportion of the explained variance amounts to 12% (0.09 + 0.03), so this plot provides only a correspondingly limited representation of the data. The same concern applies to the representation of the relation pairs in Fig. 2, which are rarely parallel nor have the same length. Nevertheless, some analogies present a clearly square-like shape, as noted in several articles (Mikolov et al., 2013c). We also plotted the same analogies on the third and fourth components (PC3 and PC4), revealing other shapes, where some are also square. This indicates that many more principal components than just the first two would be necessary in order to obtain a faithful representation of the word analogies as well as the semantic relations.

By contrast, we also performed a Principal Component analysis of Tab. 2, i.e. the **WordNet vectors** fitted by *3CosAvg*, and plotted the two first components in Euclidean space. (Fig. 3). The proportion of variance explained by each Principal Component (PC) is reported in the parentheses, and we see that these two-dimensional plots provide a very reasonable representation of the 300-dimensional vectors, since they explain a large part of the overall variance (35%+31% for Skip-gram, and 64%+17% for GloVe). In fact, with Glove, the majority of the variance is already explained by the first PC, which is very close to the axis formed by the hypernymy and hyponymy vectors. The overall structure of both models is essentially similar: in both cases the hypernym vector is the exact opposite of the hyponym vector. Also, in both cases, the antonym and synonym vectors are very close to the center, which is not surprising since the theory predicts that *3CosAvg* should yield only zero for all the parameters of symmetric relations.

3.2 Performance

The percentages shown in Tab. 3 are even numbers, because each test consists in fifty questions, so we measure *precision* by simply doubling the number of correct answers, which is a whole number between zero and fifty. A correct answer means that the best ranking prediction is a member of the set of accepted answers.

Overall, the Glove model outperformed Skip-gram with almost all relations and methods. We observe that both the random (*rnd.*) and the deterministic (*det.*) variants of *LRCos* outperform *3CosAvg* (*weight=1*) by a wide margin, while the latter only slightly improves on the *Only-B* (*weight=0*) baseline. But increasing the weight in *3CosWeight* improved the results for all *asymmetric* relations in both models: higher weights (like 3, 4 and 5) thus clearly improved over *3CosAvg*, while reducing the distance to LRCos. Moreover, *3CosWeight* provided the best results for *hyponymy* completion with both Skip-gram and Glove, and the best results for all the three kinds of *meronymy* overall. However, the optimal weight differs for each relation, suggesting a need for more research, in order to explain these variations.

Previous overall precision for the same *Lexi-*

cography tests and *3CosAvg* was 13% with GloVe and 9.6% with Skip-gram, while *LRCos*, also then, showed clearly superior performance, with 16.8% and 15.4% respectively (Gladkova et al., 2016). These results cannot be directly compared with ours, since they were obtained with other embeddings, but they show the same main trends, especially concerning the superiority of Glove over Skip-gram and of LRCos over 3CosAvg.

A striking observation is that the performance curve is completely flat across all the deterministic methods, applied to the symmetric relations (antonymy and synonymy). In this case, neither *3CosAvg* nor the deterministic *LRCos* can improve on the *Only-B* baseline, although the random variant of *LRCos* shows small occasional improvements or degradations obtained by chance, and thus unlikely to be consistently reproducible or predictive of performance on downstream tasks.

4 Discussion

4.1 Symmetry and asymmetry

Our results confirmed that *symmetry* and *asymmetry* are important mathematical properties of some WordNet relations, which determine the performance of the classification methods used in this study. *Synonymy* and *antonomy* are perfectly symmetric relations in WordNet, since every (a,a') pair is reversible, so the a class is identical to the a' class. Hence, their class-wise averages are also identical, and the difference of both averages is zero in theory, though in practice floating-point arithmetics represent the result as a very small number (cf. Tab. 2). For this reason, the *3CosAvg* method actually reduces to *Only-B*, when applied to symmetric WordNet relations. In the BATS, the same relations are not symmetric, which explains why results obtained by training on BATS alone are unlikely to transfer well to downstream tasks. Likewise, when the symmetry is lost due to implementation heuristics, the result cannot be expected to adequately handle real-world data.

With *asymmetric* relations, the set of source words may overlap to some extent with the set of target words. In particular, many words have both *hypernyms* and *hyponyms*, and contribute to the average of both classes. So, for these relations, the class-wise difference of averages only stems from the top and leaf words in the relation graph.

4.2 Polysemy

WordNet 3.1 distinguishes between thirteen senses of *man*, two of which are *antonyms* of two senses out of the four senses of *woman*, while one of the ten senses of *king* ("a male sovereign") is an antonym of *queen* ("A female sovereign ruler"), though in another sense ("a competitor who holds a preeminent position"), *king* and *queen* are synonyms.

Standard word embeddings express all the different senses of the same word with only one vector, but use different vectors for each morphological form of the same lemma. On the contrary, WordNet collapses the different word forms into one lemma, but distinguishes between the various senses of each word. Thus, WordNet fits with the word embeddings through the particular word forms, which correspond to only one morphological variant of their lemma, but aggregate all of its senses indiscriminately.

This structural discrepancy between both word models may be a major reason for the relatively low performance of standard word embeddings on *lexicographic semantics* tasks. Then it should be possible to obtain better results with lemma-based embeddings, and even better performance could be expected from word-sense vectors (Arora et al., 2018).

4.3 Future Work

The retrofitting of embeddings to semantic relations (Faruqui et al., 2014) is compatible with our method, because it is possible to fit relations to embeddings that were retrofitted to the same relations. However, we do not know if the respective benefits of both approaches could accumulate. Retrofitting brings related vectors closer together, and thus further apart from unseen words, although these could potentially be related as well, in which case we may suspect that the downstream performance actually could degrade.

A more promising approach consists in pursuing three distinct optimization goals simultaneously (Bouraoui et al., 2018): the *(source, target)* pair should belong to the given relation, while the *source* word should be a member of the *source* class (in analogies this is already known), and the *target* word be a member of the *target* class. *3CosAvg* tests the first goal, the second is always true in analogy completion tasks, and *LRCos* tests the third. Combining these objectives has been

shown beneficial with the BATS relations as training set (Bouraoui et al., 2018).

However, the BATS relations do not provide enough examples to train a classifier that can generalize adequately to downstream tasks. In particular, the lack of symmetry in the BATS synonyms and antonyms does not allow to recognize important mathematical properties of these relations. More semantic tests are needed, and the BATS is still too small. Larger tests derived from WordNet itself seem promising (Piasecki et al., 2018), though these would be limited to the word pairs known in WordNet, resulting in a limited ability to predict the performance on related pairs outside WordNet.

More successful detection of hypernyms and meronyms has been achieved using k-means clustering with the Polish wordnet (Czachor et al., 2018), so for these relations it might be possible to improve our results with similar techniques. In particular, the present study does not include *indirect relations*, although augmenting the hypernym training set with the *transitive hypernyms* would very probably be an advantage, since the BATS answer sets includes them.

5 Conclusion

We fitted WordNet relations to word embeddings, using *3CosWeight*, a new, *weighted* variant of *3CosAvg*, which allows to emulate well-known methods like *3CosAvg*, *Only-B* and *Add-Opposite*.

We showed that none of the tested classifiers can learn to distinguish between source and target classes of symmetric relations like *synonymy* and *antonymy*, since these classes are identical.

This study confirmed the superiority of *LRCos* over *3CosAvg* for learning *hyperonymy*, while *3CosWeight* was more successful with *hyponymy* and *meronymy*, suggesting that *3CosWeight* can provide a useful alternative to the other methods.

Still, the performance of these methods remains modest, and might eventually benefit from being applied to semantically disambiguated word-sense embeddings, or combined with complementary approaches.

Acknowledgments

Thanks to the anonymous reviewers and the participants at GWC 2019 in Wroclaw, and in particular to Christiane Fellbaum, Hugo Gonçalo Oliveira and Maciej Piasecki for their insightful comments and suggestions, which helped to improve this article.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. 2018. Relation induction in word embeddings revisited. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1627–1637, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Gabriela Czachor, Maciej Piasecki, and Arkadiusz Janz. 2018. Recognition of hyponymy and meronymy relations in word embeddings for polish. In *Proceedings of the 9th Global WordNet Conference (GWC 2018)*, page 254.

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.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv:1411.4166*.

Christiane Fellbaum. 1998. *WordNet, An Electronic Lexical Database*. MIT Press, Cambridge.

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 Student Research Workshop*, pages 8–15.

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*, pages 40–45. Association for Computational Linguistics.

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.

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

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv:1309.4168*.

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*, pages 3111–3119.

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*, pages 746–751.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Maciej Piasecki, Gabriela Czachor, Arkadiusz Janz, Dominik Kaszewski, and Paweł Kedzia. 2018. Wordnet-based evaluation of large distributional models for polish. In *Proceedings of the 9th Global Wordnet Conference (GWC 2018)*.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148.

Natalie Schluter. 2018. The word analogy testing caveat. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 242–246, June.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 545–550.