Specializing Word Vectors by Spectral Decomposition on Heterogeneously Twisted Graphs

Yuanhang Ren[1] **and Ye Du**[2]
[1]University of Electronic Science and Technology of China
[2]Southwestern University of Finance and Economics, China
{ryuanhang,henry.duye}@gmail.com

Abstract

Traditional word vectors, such as word2vec and glove, have a well-known inclination to conflate the semantic similarity with other semantic relations. A retrofitting procedure may be needed to solve this issue. In this work, we propose a new retrofitting method called Heterogeneously Retrofitted Spectral Word Embedding. It heterogeneously twists the similarity matrix of word pairs with lexical constraints. A new set of word vectors is generated by a spectral decomposition of the similarity matrix, which has a linear algebraic analytic form. Our method has a competitive performance compared with the state-of-the-art retrofitting method such as AR (Mrkšić et al., 2017). In addition, since our embedding has a clear linear algebraic relationship with the similarity matrix, we carefully study the contribution of each component in our model. Last but not least, our method is very efficient to execute[1].

1 Introduction

Word embedding is one of the core research areas in natural language processing. Its usefulness has been demonstrated in a wide variety of NLP tasks, e.g. dependency parsing, sentiment analysis, and machine reading comprehension. Modern embedding methods are usually based on the distributional hypothesis, namely, co-occurred words tend to purport similar semantic meanings. Although the distributional word vectors perform well on lots of tasks, they have a well-known tendency to conflate the semantic similarity information with the semantic relatedness (Hill et al., 2015). Therefore, the similarity between word vectors cannot reflect the precise semantic relation between word pairs, but just a semantic association (Yih et al., 2012). For instance, if two words are antonyms, their corresponding word vectors could be very close geometrically, which makes it very hard to distinguish one word from the other.

One way to solve this problem is to inject some lexical constraints, such as antonym relationships into the word vectors, the aim is to make antonyms far apart from each other. This process is often referred as *semantic specialization* (Mrkšić et al., 2017). There are two kinds of semantic specialization methods: (1) *joint specialization* methods, in which word vectors are trained from scratch by incorporating the lexical knowledge into the learning objective of the distributional models. Pham et al. (2015) inject a synonym/antonym margin loss into the skip-gram objective to enforce that antonym pairs have low similarity while synonym pairs have high similarity; (2) *retrofitting* methods (also referred as *post-processing methods*), in which pre-trained word vectors are fine-tuned by injecting the lexical information into vector spaces. Mrkšić et al. (2017) proposed the ATTRACT-REPEL(AR) algorithm which tries to push or pull a pair of words by a margin compared with its corresponding negative samples. Another retrofitting method is to inject lexical constraints into the word similarity matrix and obtain the tuned word vectors via the matrix decomposition. The method proposed by Sedoc et al. (2017) is in this line of research. Generally speaking, the retrofitting methods have a better performance (Mrkšic et al., 2016) while the joint specialization methods can specialize all words.

In this paper, we want to address the following question: *Can we design a retrofitting method that is interpretable and has a competitive performance with the start-of-the-art methods?* Thus, we propose a

[1]We release our code and relevant datasets at https://github.com/ryh95/HRSWE.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3599–3609
Barcelona, Spain (Online), December 8-13, 2020

novel model called HRSWE (**H**eterogeneously **R**etrofitted **S**pectral **W**ord **E**mbedding). The basic idea is that if the difference between the similarity of an antonym pair and the minimum similarity of all pairs is large (small), the weight of lexical constraint about this antonym pair will be high (low). Similar ideas are applied for synonym pairs. Moreover, words i and k will tend to be synonyms if words i and k have a common synonym(antonym) j. On the other hand, i and k will tend to be antonyms if (i, j) are synonyms and (j, k) are antonyms or the other way around. This phenomenon is called *contagion*, which will be modeled via the matrix multiplication of thesauri matrices. After the similarity matrix is constructed, we do spectral decomposition on the similarity matrix to obtain the specialized embedding. We also care about whether the performance of a specialization method still holds when the thesauri used by the method is perturbed, which is called the *robustness*. In this paper, we explore a few perturbation methods. Overall, HRSWE is slightly better than AR in terms of robustness given these perturbations. The contributions of our method are as follows.

- Foremost, our method is more interpretable than AR in three folds. First, our embedding has a clear algebraic relationship with the original word embedding, while ATTRACT-REPEL does not have this property. In particular, our embedding can be formulated as an analytical form in terms of the injected similarity matrix. Furthermore, the importance of synonym and antonym information injected into the similarity matrix is quantified by experiments. Finally, the significance of the contagion information is demonstrated by experiments as well.
- In terms of performance, on one hand, our novel method not only has a much better performance compared with Word2Vec but also achieves a competitive performance with the state-of-the-art method ATTRACT-REPEL on three tasks. Furthermore, our method has slightly better robustness compared with ATTRACT-REPEL. On the other hand, our method is faster than ATTRACT-REPEL by at least one order of magnitude in terms of running time. It makes our method appealing.

2 The Methodology

Let $V = \{v_1, v_2, ...v_n\}$ be the vocabulary set, $S = \{(v_i, v_j)|v_i \text{ is a synonym of } v_j\}$ be the synonym set, and $A = \{(v_i, v_j)|v_i \text{ is an antonym of } v_j\}$ be the antonym set, where n is the number of words. The original word vector set is $\{x_1, ..., x_n\}$, where $\forall i, x_i \in \mathbb{R}^d$. The word vectors matrix $X = [x_1|...|x_n] \in \mathbb{R}^{d\times n}$ is obtained by stacking the d dimensional original word vectors one by one horizontally. Then, the similarity matrix is defined as

$$W = X^T X.$$

Next, the synonym and antonym thesauri information S_0 and A_0 are introduced, where

$$S_0(v_i, v_j) = \begin{cases} 1 & \text{if} \quad (v_i, v_j) \in S \\ 0 & \text{otherwise} \end{cases}, A_0(v_i, v_j) = \begin{cases} 1 & \text{if} \quad (v_i, v_j) \in A \\ 0 & \text{otherwise} \end{cases}.$$

After that, we consider the thesauri contagion information which is defined in Figure 1. Two same types of relations sharing a common word produce a synonym relation. Otherwise, an antonym relation will be induced. Let the original thesauri be

$$T_0(v_i, v_j) = \begin{cases} a & \text{if} \quad (v_i, v_j) \in S \\ -b & \text{if} \quad (v_i, v_j) \in A \end{cases}$$

where a and b are two positive hyperparameters. Given the definition of the contagion, the similarity between words j and k can be modeled as

$$T_1^{j,k} = \sum_{i=1}^{n} T_0^{j,i} T_0^{i,k}$$

considering all words. This is indeed the matrix multiplication.

$$T_1 = T_0 T_0$$

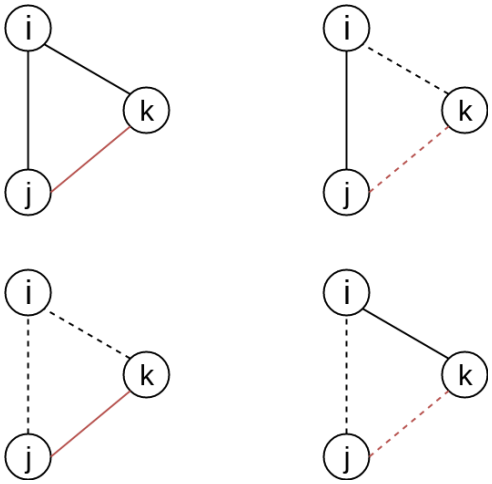

Figure 1: The definition of the thesauri contagion. Given words i, j, and k, black links are given links while red ones are predicted contagion links. Synonym relations are presented in solid lines while antonym relations are denoted as dashed lines.

Then, we extract the synonym and antonym contagion information as follows[2],

$$S_1(v_i, v_j) = \begin{cases} T_1(v_i, v_j) & \text{if} \quad T_1(v_i, v_j) > 0 \\ 0 & \text{otherwise} \end{cases}, A_1(v_i, v_j) = \begin{cases} -T_1(v_i, v_j) & \text{if} \quad T_1(v_i, v_j) < 0 \\ 0 & \text{otherwise} \end{cases}.$$

Finally, we combine S_0, A_0, S_1, and A_1 with W to obtain the thesauri injected similarity matrix $\hat{W}$,

$$\begin{aligned} \hat{W} = \beta_0 W + \beta_1(W_{max} - W) \odot S_0 - \beta_2(W - W_{min}) \odot A_0 \\ + \beta_1(W_{max} - W) \odot S_1 - \beta_2(W - W_{min}) \odot A_1 \end{aligned} \quad (1)$$

where $\odot$ is the Hadamard product(elementwise multiplication). The W_{max} and W_{min} are the maximum and minimum of W and are used as the similarity baselines of synonym pairs and antonym pairs to guide the thesauri injection. The β_0, β_1, and β_2 are hyperparameters. Note that the weight of lexical constraints information injected into one word pair depends on the similarity calculated by its original word vectors. For instance, the weights injected for two synonym words with lower original similarity (implied by W) would be larger compared with another synonym pair with higher original similarity. Thus, the weights are heterogeneous.

Recall that our goal is to construct d dimensional specialized word vectors $\hat{V} \in \mathbb{R}^{d \times n}$. Given $\hat{W}$, it can be achieved as follows.

$$\min_{\hat{V}} \quad \|\hat{W} - \hat{V}^T \hat{V}\|_F$$

The problem is equivalent to find a matrix $\hat{W}_{SD}$ such that

$$\begin{aligned} \min_{\hat{W}_{SD}} \quad & \|\hat{W} - \hat{W}_{SD}\|_F \\ s.t. \quad & \hat{W}_{SD} \succeq 0, \text{rank}(\hat{W}_{SD}) \leq d \end{aligned} \quad (2)$$

where the notation $\hat{W}_{SD} \succeq 0$ means that $\hat{W}_{SD}$ is symmetric and positive semidefinite(SPSD).

Since $\hat{W}$ is a symmetric matrix, it has a truncated spectral decomposition

$$\hat{W} \approx Q_d \Lambda_d Q_d^T,$$

[2]Values in T_1 are clipped to $[-1, 1]$ after the multiplication.

Algorithm 1 HRSWE

Input: the original embeddings X, the synonym thesauri S_0, the antonym thesauri A_0, and the entire thesauri T_0
Parameter: hyperparameters $\beta_0, \beta_1, \beta_2, a, b$
Output: the specialized embeddings $\hat{V}$

1: Denote $W = X^T X$.
2: Let $T_1 = T_0 T_0$ and clip values in T_1 to $[-1, 1]$.
3: Extract S_1 and A_1 from T_1.
4: Combine S_0, A_0, S_1, A_1, and W to obtain $\hat{W}$ according to equation (1).
5: Do a truncated spectral decomposition on $\hat{W}$ to obtain $\hat{V}$, $\hat{V} = \Lambda_d'^{\frac{1}{2}} Q_d^T$.
6: **return** $\hat{V}$

where Λ_d is a diagonal matrix containing the largest d eigenvalues with respect to multiplicities, Q_d is the d eigenvectors corresponding to the largest eigenvalues of $\hat{W}$. According to Dax (2014), there is an analytic optimal solution to this nearest low-rank SPSD matrix problem

$$\hat{W}_{SD} = Q_d \Lambda_d' Q_d^T,$$

where Λ_d' is obtained by replacing the negative values of Λ_d with 0. Thus, the $\hat{V}$ is

$$\hat{V} = \Lambda_d'^{\frac{1}{2}} Q_d^T$$

The time complexity of this method is $O(n^2 d)$ and the method is summarized in Algorithm 1. For a large n, one can use matrix sparsification methods, Nyström methods, and GPUs to accelerate the eigendecomposition.

3 Experiments

In this section, we evaluate the methods on four tasks: the word similarity, the synonymy/antonymy classification, the lexical simplification, and the robustness test.

Basic Setup To evaluate the effectiveness of our method, Word2Vec (Mikolov et al., 2013) is our original embedding. Specifically, we choose the 300-dimensional skip-gram vectors[3] and denote this original embedding as SGNS-GN. The synonym and antonym relationships in Vulić (2018) are adopted as the lexical constraints. We call this set Ω. Throughout the paper, the main baseline of our method is AR. Although the method proposed by Sedoc et al. (2017) is sensitive to thesauri according to our private communication, we try their homogeneous graph construction method and do spectral decomposition on this graph. This benchmark is called RSWE. All our experiments are carried out on a server with an NVIDIA RTX 2080 Ti GPU and an Intel i9-7940x CPU with 32 GB of RAM.

3.1 Word Similarity

The word similarity task is a standard evaluation task for word embeddings. There are several datasets containing pairs of words that their semantic similarities are labeled by humans. On the other hand, the similarity of a pair of words can be predicted by our word embeddings, e.g. the cosine similarity between the word vectors of the pair. By computing the Spearman's ρ rank correlation between human's scores and our predictions, one can measure how well word embedding models the semantic similarities. We evaluate our methods with two recent datasets: SimLex-999 (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016). In the following, we will describe how we validate and test those models.

For HRSWE, the vocabulary set V is composed of all the words in SimLex-999 and SimVerb-3500. Meanwhile, the lexical constraint sets S and A are constructed in the following way: if $v_i, v_j \in V$, $v_i \neq v_j$ and the pair (v_i, v_j) belongs to our constraint set Ω, (v_i, v_j) will be added to S or A correspondingly. We don't have to train these two models. The only thing to do is to choose the hyperparameters β_0, β_1, β_2, a, and b. To quantify how different information affects the quality of the specialized embedding, we also evaluate two degenerate cases of HRSWE(called HRSWE-1/HRSWE-2, the complete

[3]Available at `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing`

model is HRSWE-3). They both make no use of the contagion information and HRSWE-1 just uses one hyperparameter in which $\beta_0 = 1$ and $\beta_1 = \beta_2$. RSWE-2 and RSWE-3 are similar with HRSWE-2 and HRSWE-3 except that $W_{max} - W$ and $W - W_{min}$ are replaced with W. For the state-of-the-art retrofitting model ATTRACT-REPEL, it is trained on the same vocabulary V and lexical constraint sets S and A as our method. This is quite different from its original implementation in Vulić (2018), where its lexical constraint set is Ω. This makes our implementation more like a customization method, in which only the set of words and constraints related to the task will be used. In terms of the hyperparameters selection, all HRSWE hyperparameters take values from $[0, 1]$ and the same ranges will be applied for RSWE. The hyperparameters ranges of AR are $[0, 1]$, $[0, 1]$, $\{64, 128, 256\}$, $\{1, 2, ..., 20\}$, $[10^{-9}, 10^{0}]$ for the synonym margin, antonym margin, mini-batch size, number of epochs, and regularization strength respectively. Bayesian optimization with Gaussian Processes is employed to search hyperparameters for all methods. All models are tuned on the validation set of the SimVerb-3500.

MODEL	SimLex	SimVerb-test	Specializing Time(s)
SGNS-GN	44.2	35.8	/
ATTRACT-REPEL	**77.6**	74.3	38.5
HRSWE-1 (β_0=1,$\beta_1 = \beta_2$,w/o T_1)	73.5	71.9	**0.7**
HRSWE-2 (w/o T_1)	73.7	73.1	0.8
RSWE-2 (w/o T_1)	74.2	69.8	**0.7**
HRSWE-3	76.0	**75.7**	0.9
RSWE-3	75.5	71.2	0.9

Table 1: **Word Similarity task.** Results of different models on two word similarity datasets (Spearman's ρ).

The test results are summarized in Table 1. First, let's focus on the quality of HRSWEs. The HRSWE-3 has a comparable performance with AR. On the SimVerb, it outperforms AR by about 1 point while it is slightly worse than AR by 1-2 points on the SimLex. Furthermore, our experiments demonstrate why our method can compete with the state-of-the-art method in an ablation fashion. Very intuitively, the number of hyperparameters is important. For instance, HRSWE-2 has three hyperparameters while HRSWE-1 has only one. Given the more flexibility to tune the weights of the original, synonym, and antonym information, HRSWE-2 yields a performance boost compared with HRSWE-1. The corresponding values of hyperparameters are $\beta_0 = 0.70$, $\beta_1 = 0.29$, and $\beta_2 = 1.0$. It shows that the performance boost comes from decreasing the contribution of the original information and increasing the contribution of the antonym information. Most importantly, the contagion information is a necessity under our setting. With the aid of that, the performance of HRSWE-3 exceeds that of HRSWE-2 by about 2 points, which makes our method even on a par with AR[4]. The performance of RSWE is much poorer than HRSWE which demonstrates the advantages of the heterogeneous twisting. Next, we put our attention on the efficiency of the methods. Given V, X, A, S, the search ranges of hyperparameters, and the number of search rounds, the average computation time to generate the specialized embeddings is documented, which is denoted as the specializing time. Our method is much more efficient than AR. The specializing time of HRSWE is about 0.8s which is nearly 50 times faster than that of AR. Note that in all our implementations of ATTRACT-REPEL, the GPU is utilized.

3.2 Synonym/Antonym Classification

The synonym/antonym classification task is a binary classification task to decide whether pairs of words are synonyms or antonyms. Given the word embeddings of a pair of words and a threshold γ, if the cosine similarity of the pair is higher than the given threshold, this pair is regarded as synonyms; otherwise, they

[4]We also extract the synonym and antonym contagion pairs from HRSWE-3 to augment the training data for AR. However, the performance of AR deteriorates under this setting.

are antonyms. We evaluate our methods on a recent dataset proposed by Nguyen et al. (2017b). It has a validation set and a test set. Both sets consist of noun pairs, verb pairs, and adjective pairs.

For HRSWE, the vocabulary set V consists of all words in the validation set and test set of Nguyen et al. (2017b). The lexical constraints are constructed in the same way as that in the word similarity task. For ATTRACT-REPEL, the vocabulary and the lexical constraints are the same as our methods. Hyperparameters ranges of HRSWE are $\beta_0 \in [0, 1]$, $\beta_1 \in [10^{-1}, 10^1]$, $\beta_2 \in [10^{-1}, 10^1]$, $a \in [0, 1]$, and $b \in [0, 1]$. The same ranges will be applied for RSWE. The hyperparameters ranges of AR are the same as those in the word similarity task.

MODEL	A	V	N	Specializing Time(s)
SGNS-GN	66.7	66.7	66.7	/
ATTRACT-REPEL	**97.5**	96.9	**86.0**	311.8
HRSWE-1 (β_0=1,$\beta_1 = \beta_2$,w/o T_1)	94.8	91.1	80.2	44.0
HRSWE-2 (w/o T_1)	95.7	94.5	84.0	**42.4**
RSWE-2 (w/o T_1)	93.2	91.1	82.8	44.7
HRSWE-3	97.0	**97.0**	**86.0**	42.7
RSWE-3	93.2	91.1	82.3	43.7

Table 2: **Synonym/Antonym Classification task.** Results (F_1) of different models on Adjective, Verb and Noun test pairs.

The test results are listed in Table 2. From the specialization quality perspective, HRSWE-3 is competitive with AR. The F_1 difference between the two models is quite small, which is within 1 point. In the meantime, we analyze the components of HRSWE, answering why it specializes embeddings so well. Similar to the word similarity tasks, more hyperparameters will enhance performance. For instance, HRSWE-2 improves the performance on Verb and Noun by 3-4 points compared with HRSWE-1. The corresponding βs are $\beta_0 = 0.0$, $\beta_1 = 1.87$, and $\beta_2 = 7.25$. Surprisingly, β_0 is 0.0 in this test. Note that 85% percent of the task tuples are covered by thesauri tuples. This might be one reason for that. Meanwhile, the contagion information is also crucial to this task. Without that, HRSWE-3 cannot surpass the HRSWE-2 by 2 points. On the other hand, the F_1 score of HRSWE is higher than RSWE by about 3 points which accentuates the need for the heterogeneous twisting. From the efficiency perspective, the specializing time of HRSWE is significantly less than that of AR by around 7-8 times.

3.3 Lexical Simplification

We now evaluate HRSWE on a downstream task called Lexical Simplification. The goal of this task is to replace the complex words that are used less frequently and known to fewer speakers with their simpler and frequently used synonyms. For instance, given a sentence "the notorious pirate won the match", one may expect the word "notorious" to be replaced by some other simpler words like "infamous". We choose the dataset crowdsourced by Horn et al. (2014) as the task data. It contains 500 sentences and each of the sentences has one target word. For each target word, it has a candidate set. Simplification models are expected to replace the target words with words or phrases that in candidate sets. The 500 sentences are equally split into validation and testing sets. The LIGHT-LS model (Glavaš and Štajner, 2015) is adopted as the simplification model[5].

For HRSWE, the vocabulary V is prepared as follows. First, we exclude phrases in all candidate sets. Since the LIGHT-LS retrieves simpler words from the embedding space and lots of phrases are not in the space, the phrases in candidate sets will not be retrieved and are removed from our vocabulary. Second, we lemmatize all the target words and words in the rest of the candidate sets as the lemmatized words will be found in constraints more easily. Finally, the lemmatized words and words in sentences will be added to V. The lexical constraints are constructed in the same way as that in the word similarity

[5]Available at `https://github.com/codogogo/lightls`

task. For ATTRACT-REPEL, the vocabulary and the lexical constraints are the same as our methods. The hyperparameter range of HRSWE is the same as the synonym/antonym classification task. Hyperparameters ranges of ATTRACT-REPEL are the same as the word similarity task except that the range of the mini-batch size is $\{32, 64, 128, 256, 512, 1024\}$ and the range of the number of epochs is $[1, 15]$.

MODEL	Accuracy	Specializing Time(s)
SGNS-GN	42.7	/
ATTRACT-REPEL	57.6	72.3
HRSWE-1 (β_0=1,$\beta_1 = \beta_2$,w/o T_1)	58.5	2.6
HRSWE-2 (w/o T_1)	**58.7**	**2.6**
HRSWE-3	57.5	2.9

Table 3: **Lexical Simplification task.** Results of AR and HRSWE on the Horn's Lexical Simplification dataset.

The test results are listed in Table 3. From the specialization quality perspective, HRSWE-2 surpasses AR by about 1 point. This might because fewer hyperparameters can be better tuned given the same number of hyperparameter search rounds[6]. From the efficiency perspective, HRSWE is about 25 times faster than AR. To summarize, the HRSWE has a competitive specialization quality compared with AR while runs dramatically faster.

3.4 The Robustness Test

Can specialization methods still produce high-quality embeddings given the perturbation in the thesauri? In this section, we evaluate the robustness of our method and AR on the word similarity task and the synonym/antonym classification task given three types of perturbed thesauri. The perturbation methods are described as follows.

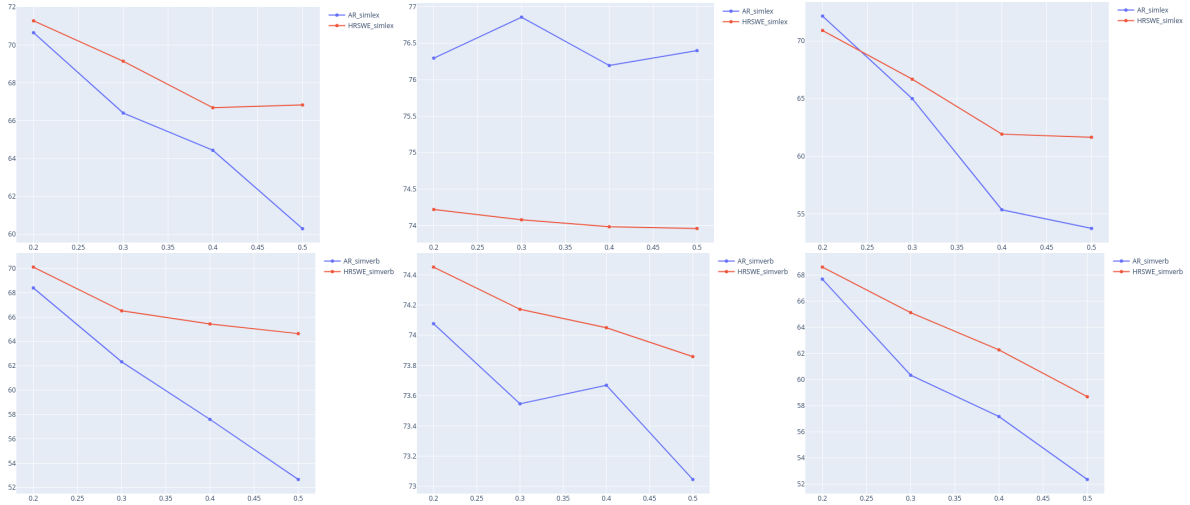

Figure 2: Results(Spearman's ρ) of HRSWE and AR on two word similarity datasets with respect to three types of perturbations. The first and second rows represent results on the SimLex and SimVerb-test datasets. The first, second, and third columns represent results with Syn-Adv, Ant-Adv, and Syn-Ant-Adv perturbations respectively. The red lines are results of HRSWE. The x-axis is the proportion of the subset in the intersection r.

[6]Note that our accuracies are much lower than those in Table 4 of Glavaš and Vulić (2018), we believe the reason is that our "accuracy" is different from theirs. Our accuracy is the number of correct simplifications divided by the number of test sentences while the denominator of the accuracy in Glavaš and Vulić (2018) is the number of "indicated complex words".

First, we extract word pairs from the validation and test sets in a particular task and create a union set from both sets. Second, the union set is intersected with the synonym part of the thesauri. Finally, a random subset of the intersection is moved into the antonym part of the thesauri. The proportion of the subset in the intersection is denoted as r. This perturbation method is called **Syn-Adv** and one can also put the antonym intersection into the synonym thesauri which is denoted as **Ant-Adv**. By combining the **Syn-Adv** and **Ant-Adv**, we obtain the **Syn-Ant-Adv** perturbation.

We first evaluate the robustness of HRSWE and AR on word similarity tasks. The initial V, A, and S are constructed in the same way as the previous word similarity task. The hyperparameter ranges of HRSWE and AR are almost the same as the previous word similarity task except that the range of the mini-batch size is $\{32, 64, 128, 256, 512, 1024\}$. The proportion r varies from 0.2 to 0.5. The test results are presented in Figure 2. Interestingly, HRSWE achieves a better performance in most cases(5 cases out of 6) over 3 types of perturbation. Given the perturbation of Syn-Adv and Syn-Ant-Adv, HRSWE outperforms AR by 0.6-12 points on SimLex and SimVerb-test. As the proportion increases, the performance gap between the two methods increases. Nonetheless, AR wins the round on the SimLex perturbed via Ant-Adv by around 2 points on average. This phenomenon might be related to the fact that the set of antonym intersection takes only about 8% of SimLex word pairs, which is too low.

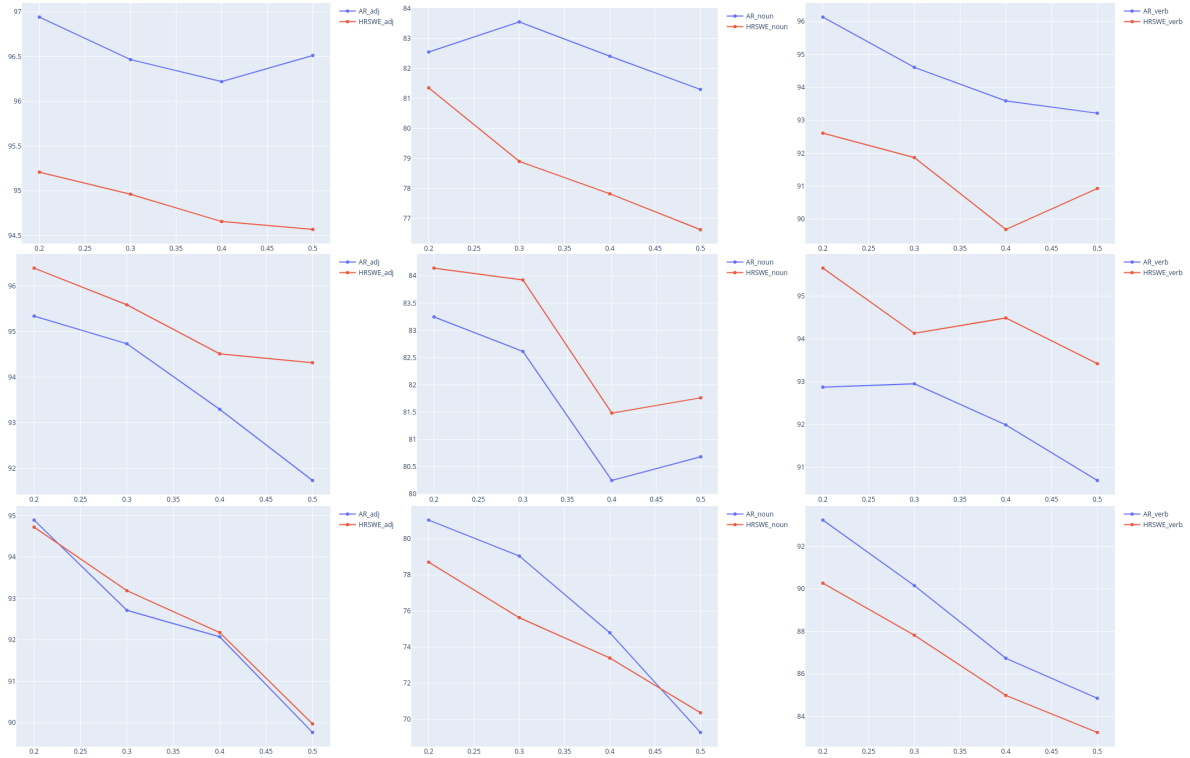

Figure 3: Results (F_1) of HRSWE and AR on Adjective, Verb and Noun test pairs with respect to three types of perturbations. Rows from top to bottom represent the results with Syn-Adv, Ant-Adv, and Syn-Ant-Adv perturbations. Columns from left to right represent the results on Adjective, Noun, and Verb test pairs. The red lines are results of HRSWE. The x-axis is the proportion of the subset in the intersection r.

We then evaluate the robustness of HRSWE and AR on the synonym/antonym classification task. The hyperparameter ranges of AR are the same as those in the word similarity perturbation tasks while the ranges of HRSWE are almost the same as those in the previous synonym/antonym classification tasks except replacing the $\beta_1 \in [10^{-1}, 10^1]$ and $\beta_2 \in [10^{-1}, 10^1]$ with $\beta_1 \in [0, 1]$ and $\beta_2 \in [0, 1]$ in the Syn-Adv perturbation. The test results are demonstrated in Figure 3. Overall, HRSWE is slightly worse than AR. On average, AR surpasses HRSWE by about 2.9 points on all the three datasets perturbed by Syn-Adv while it falls behind our method by about 1.6 points on all the three datasets perturbed by Ant-

Adv. For the Syn-Ant-Adv perturbation, the two methods are almost on par on Adjective pairs while AR is slightly better than HRSWE on Noun and Verb pairs.

Summarization and Discussion In all, HRSWE beats AR on 9 robustness test cases out of 15 and wins 92 points in total while loses 58 points. Among all rs in all test cases, the performance of HRSWE is better than that of AR up to 12 points when r is 0.5 with the Syn-Adv perturbation on the SimVerb-test dataset. On the other hand, AR exceeds HRSWE by up to 5 points in the Syn-Adv perturbation on the Noun dataset when r is 0.5, which is the best scenario that AR outperforms HRSWE. From these perspectives, we argue AR is slightly less robust than our method.

To reduce the impact of the perturbation, the thesauri contagion information should be fully exploited. The usage of the contagion information in our method is explicit while in AR is *implicit*. Suppose we have three words i, j, k, (i, j) and (j, k) are two pairs of synonyms, AR forces (i, j) to be closer than (j, k) while also forces (j, k) to be closer than (i, j). Thus, the two words i and k will probably have a high similarity in the final embedding as well. This explains why AR is resistant to the perturbations.

4 Related Works

Apart from intrinsic tasks, lots of extrinsic downstream applications would be influenced without the semantic specialization. For sentiment analysis, if "good" and "bad" are similar to each other, it would be hard to distinguish the sentiment polarity of a sentence. For spoken language understanding, it would be annoying if a user wants a cheap restaurant while the virtual assistant recommends an expensive one.

Before AR, several important post-processing methods need to be mentioned. The first post-processing method is Retrofitting (Faruqui et al., 2014) in which the word "retrofitting" is first used. After that, the method PARAGRAM (Wieting et al., 2015) extends Retrofitting with a more sophisticated "ATTRACT" term. Note that both Retrofitting and PARAGRAM do not consider antonymy, Counter-Fitting (Mrkšic et al., 2016) models both synonym and antonym relations. The differences among these methods are reviewed in Glavaś et al. (2019).

To make the retrofitting models able to specialize the entire vocabulary, Glavaš and Vulić (2018) try to explicitly retrofit the word embeddings. The basic idea is to learn a global retrofitting function using linguistic constraints as training examples. After that, Vulić et al. (2018) propose a post-specialization model that tries to train a neural network that can mimic the specialization of AR. This method yields considerable gains on a variety of tasks.

So far, the semantic similarity is a symmetric relation. Some salient asymmetric relations like hypernym and meronym also need to be modeled in the word embeddings. HyperVec (Nguyen et al., 2017a) model is a joint model that augments the skip-gram objective with the hypernym constraints and it can also tell which word is the hypernym. Vulić and Mrkšić (2017) propose a retrofitting model that extends the AR to the lexical entailment by adding the attract objective according to hypernym constraints and the asymmetric norm-based objective.

5 Conclusion

In this paper, we propose a new retrofitting method called Heterogeneously Retrofitted Spectral Word Embedding. This method shows comparable performance with the state-of-the-art retrofitting method while it is quite efficient. Besides that, our method is slightly more robust than AR under several perturbations. One major advantage of our method is its interpretability. Our specialized embedding has a clear linear algebraic relationship with original embeddings. Moreover, the impact of hyperparameters and contagion information on HRSWE has been carefully analyzed. It demonstrates the enhancement of the performance of embeddings in a step by step fashion. In the future, we would like to extend our methods to other types of word relations such as hyponymy, meronymy, and so on.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China grants 11501464 and 11761141007.

References

Achiya Dax. 2014. Low-rank positive approximants of symmetric matrices. *Advances in Linear Algebra & Matrix Theory*, 4(03):172.

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

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint arXiv:1608.00869*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Goran Glavaš and Ivan Vulić. 2018. Explicit retrofitting of distributional word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–45.

Goran Glavaś, Edoardo Maria Ponti, and Ivan Vulić. 2019. Semantic specialization of distributional word vectors. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China, November. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT*, pages 142–148.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017a. Hierarchical embeddings for hypernymy detection and directionality. *arXiv preprint arXiv:1707.07273*.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017b. Distinguishing antonyms and synonyms in a pattern-based neural network. *arXiv preprint arXiv:1701.02962*.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China, July. Association for Computational Linguistics.

Joao Sedoc, Jean Gallier, Dean Foster, and Lyle Ungar. 2017. Semantic word clusters using signed spectral clustering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 939–949.

Ivan Vulić and Nikola Mrkšić. 2017. Specialising word vectors for lexical entailment. *arXiv preprint arXiv:1710.06371*.

Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Post-specialisation: Retrofitting vectors of words unseen in lexical resources. *arXiv preprint arXiv:1805.03228*.

Ivan Vulić. 2018. Injecting lexical contrast into word vectors by guiding vector space specialisation. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 137–143.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Wen-tau Yih, Geoffrey Zweig, and John C Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222. Association for Computational Linguistics.