Geometry and analogies: a study and propagation method for word Representations

Sammy Khalife, Leo Liberti, and Michalis Vazirgiannis

LIX, Ecole Polytechnique, 91128 Palaiseau, France
{khalife,liberti,mvazirg}@lix.polytechnique.fr

Abstract. In this paper we discuss the well-known claim that language analogies yield almost parallel vector differences in word embeddings. On the one hand, we show that this property, while it does hold for a handful of cases, fails to hold in general especially in high dimension, using the best known publicly available word embeddings. On the other hand, we show that this property is not crucial for basic natural language processing tasks such as text classification. We achieve this by a simple algorithm which yields updated word embeddings where this property holds: we show that in these word representations, text classification tasks have about the same performance.

1 Introduction

1.1 Context and motivations

The motivation to build word representations as vectors in a Euclidean space is twofold. First, geometrical representations can possibly enhance our understanding of a language. Second, these representations can be useful for information retrieval on large datasets, for which semantic operations become algebraic operations. First attempts to model natural language using simple vector space models go back to the 1970s, namely Index terms [20], term frequency inverse document frequency (TF-IDF) [18], and corresponding software solutions SMART [19], Lucene [9]. In recent work about word representations, it has been emphasized that many analogies such as *king* is to *man* what *queen* is to *woman*, yielded almost parallel difference vectors in the space of the two most significant coordinates [13,16], that is to say (if $d = 2$):

$$\begin{aligned} (u_i \mid 1 \leq i \leq n) \in \mathbb{R}^d \quad & \text{being the word representations} \\ \text{(3,4) is an analogy of (1,2)} \Leftrightarrow \exists \epsilon \in \mathbb{R}^d \ \text{ s.t } \quad & u_2 - u_1 = u_4 - u_3 + \epsilon \\ \text{where} \quad & ||\epsilon|| \ll \min(||u_2 - u_1||, ||u_4 - u_3||) \end{aligned} \tag{1}$$

In Eq. (1) $||x|| \ll ||y||$ means in practice that $||x||$ is much smaller than $||y||$. Eq. (1) is stricter than just parallelism, but we adopt this version because it corresponds to the version the scientific press has amplified in such a way

that now it appears to be part of layman knowledge about word representations [12,21,4]. We hope that our paper will help clear a misinterpretation.

Recent work leads us to cast word representations into two families: *static representations*, where each word of the language is associated to a unique element (scope of this paper), and *dynamic representations*, where the entity representating each word may change on the context (we do not consider this case in this paper).

1.2 Contributions

The attention devoted in the literature and the press to Eq. (1) might have been excessive, based on the following criteria:

- The proportion of analogies leading to close parallelism is small.
- The classification of analogies based on parallelism does not appear as an easy task.

Second, we present a very simple propagation method in the graph of analogies, enabling parallelism for analogies. Our code is available online.[1]

2 Related work

2.1 Word embeddings

In the *static representations* family, after the first vector space models (Index terms, TF-IDF, see SMART [19], Lucene [9]), Skip-gram and statistical log-bilinear regression models became very popular. The most famous are Glove [16], Word2vec [13], and fastText [3]. Since word embeddings are computed once and for all for a given string, this causes polysemy for fixed embeddings. To overcome this issue, the family of *dynamic representations* have gained in attention very recently due to the increase of deep learning methods. ELmo [17], and Bert [8] representations take in account context, letters, and n-grams of each word. We do not address comparison with these methods in this paper because of the lack of analysis of their geometric properties.

There have been attempts to evaluate the semantic quality of word embeddings [10], namely:

- Semantic similarity (Calculate Spearman correlation between cosine similarity of the model and human rated similarity of word pairs)
- Semantic analogy (Analogy prediction accuracy)
- Text categorisation (Purity measure)

However, in practice, these semantic quality measures are not preferred for applications: the quality of word embeddings is evaluated on very specific tasks, such as text classification or named entity recognition. In addition, recent work

[1] Link to repository

[15] has shown that the use of analogies to uncover human biases should be carried out very carefully, in a fair and transparent way. For example [6] analyzed gender bias from language corpora, but balanced their results by checking against the actual distribution of jobs between genders.

2.2 Relation embeddings for named entities

An entity is a real-world object and denoted with a proper name. In the expression "Named Entity", the word "Named" aims at restricting the possible set of entities to only those for which one or many rigid designators stands for the referent. Named entities have an important role in text information retrieval [14].

For the sake of completeness, we report work on the representation of relations between entities. Indeed, an entity relation can be seen as an example of relation we consider for analogies (example: Paris is the capital of France, such as Madrid to Spain). There exist several attempts to model these relations, for example as translations [5,22], or as hyperplanes [11].

2.3 Word embeddings, linear structures and pointwise mutual information

In this subsection, we will focus on a recent analysis of pointwise mutual information, which aims at providing a piece of explanation of the linear structure for analogies [1,2]. This work provides a generative model with priors to compute closed form expressions for word statistics. The generation of sentences in a given text corpus is made under the following generative assumptions:

- ***Assumption 1***: The ensemble of word vectors consists of i.i.d samples generated by $v = s\,\hat{v}$, where $\hat{v}$ is drawn from the spherical Gaussian distribution in $\mathbb{R}^d$ and s is a random scalar with expectation $\tau = O(1)$, always upper bounded by constant $\kappa \in \mathbb{R}^+$.
- ***Assumption 2***: The text generation process is driven by a random walk of a vector, i.e if w_t the word at step t, there exists a discourse vector c_t such that $\mathsf{P}(w_t = w|c_t) \propto \exp(\langle c_t, v_w \rangle)$, and $\kappa > 0$ such that:

$$\begin{aligned} |s| &\leq \kappa \\ \mathbb{E}_{c_{t+1}}(e^{\kappa||c_{t+1}-c_t||_2}) &\leq 1 + \epsilon_1 \end{aligned} \tag{2}$$

In the following, we use the notations: $\mathsf{P}(w, w')$ is the probability that two words w and w' occur in a window of size 2, $\mathsf{P}(w)$ is the marginal probability of w. $\mathsf{PMI}(w, w^{'})$ is the pointwise mutual information between two words w and $w^{'}$ [7]. Under these conditions, we have the following result [1]:

Theorem 1.

$$\begin{aligned} \mathsf{PMI}(w, w') \triangleq \log \frac{\mathsf{P}(w, w')}{\mathsf{P}(w)\mathsf{P}(w')} &= \frac{\langle v_w, v_{w'} \rangle}{d} \pm O(\epsilon_2) \\ &\textit{with} \qquad \epsilon_2 = O(\epsilon_1) \end{aligned} \tag{3}$$

Eq. (3) shows that we should expect high cosine similarity for pointwise close terms (if ϵ_2 is negligible).

The main aspect we are interested in is the relationship between linear structures and analogies. In [1], the subject is treated with an assumption following [16], stated in Eq. (4). Let χ be any set of words, and a and b words are involved in a semantic relation $\mathcal{R}$. Then there exist two scalars $\nu_{\mathcal{R}}(\chi)$ and $\xi_{ab\mathcal{R}}(\chi)$ such that:

$$\frac{\mathsf{P}(\chi|a)}{\mathsf{P}(\chi|b)} = v_{\mathcal{R}}(\chi)\,\xi_{ab\mathcal{R}}(\chi) \tag{4}$$

We failed to fully understand the argument made in [1,16] linking word vectors to differences thereof. However, if we assume Eq. (4), by Eq. (3) we obtain the following.

Corollary 2. *Let V be the $n \times d$ matrix whose rows are the vectors of words in dimension d. Let v_a and v_b be vectors corresponding (respectively) to words a and b. Assume a and b are involved in a relation $\mathcal{R}$. Then there exists a vector $\xi'_{abR} \in \mathbb{R}^n$ such that:*

$$V(v_a - v_b) = d\log(v_{\mathcal{R}}) + \xi^{'}_{ab\mathcal{R}} \tag{5}$$

Proof. Let x a word, and a, b two words sharing a relation $\mathcal{R}$. From relation Eq. (4), composing with log

$$\log(\frac{\mathsf{P}(x|a)}{\mathsf{P}(x|b)}) = \log(v_{\mathcal{R}}(x)) + \log(\xi_{ab\mathcal{R}}(x)) \tag{6}$$

On the other hand, using Eq. (3)

$$\begin{aligned}
\log(\frac{\mathsf{P}(x|a)}{\mathsf{P}(x|b)}) &= \log(\frac{\mathsf{P}(x,a)\mathsf{P}(b)}{\mathsf{P}(x,b)\mathsf{P}(a)}) \\
&= \log(\frac{\mathsf{P}(x,a)\mathsf{P}(b)\mathsf{P}(x)}{\mathsf{P}(x,b)\mathsf{P}(a)\mathsf{P}(x)}) \\
&= \mathsf{PMI}(x,a) - \mathsf{PMI}(x,b) \\
\log(\frac{\mathsf{P}(x|a)}{\mathsf{P}(x|b)}) &= \frac{\langle v_x, v_a - v_b\rangle}{d} + \epsilon_{ab}(x)
\end{aligned} \tag{7}$$

Combining equations (6) and (7), for any x:

$$\langle v_x, v_a - v_b\rangle = d\log(v_{\mathcal{R}}(x)) + d(\log(\xi_{ab\mathcal{R}}(x)) - \epsilon_{ab}(x)) \tag{8}$$

Let V the matrix whose rows are the word vectors: $V(v_a - v_b)$ is a vector of $\mathbb{R}^n$ whose component associated with word x is exactly $\langle v_x, v_a - v_b\rangle$. Then, let v_R and $\xi^{'}_{abR}$ the vectors of components $v_R(x)$ and $d(\log \xi_{abR}(x) - \epsilon_{ab}(x))$. Then, Eq (8) is exactly Eq (5). □

It is shown in [1] that $||V^{+}\xi'_{abR}|| \leq ||\xi^{'}_{abR}||$, where V^{+} is the pseudo-inverse of V. In other words, the "noise" factor ξ' can be reduced. This reduction may not be sufficient if ξ_{abR} is too large to start with. In the next section we shall propose an empirical analysis of existing embeddings with regard to analogies and parallelism of vector differences.

3 Experiments with existing representations

In this section, we present a list of experiments we ran on the most famous word representations.

3.1 Sanity check

The exact meaning of the statement that analogies are geometrically characterized in word vectors is as follows [12,16]. For each quadruplet of words involved in an analogy (a, b, c, d), consider the word vector triplet (v_a, v_b, v_c), and the difference vector $x_{ab} = v_b - v_a$. Then we run PCA on the set of word vectors to get representations in $\mathbb{R}^2$. Find the k nearest neighbours of $v_c + x_{ab}$ in the word embedding set (with k small). Finally, examine the k words and choose the most appropriate word d for the analogy $a : b = c : d$. We ran this protocol in many dimension with a corpus of analogies. We display the results obtained in Fig. 1.

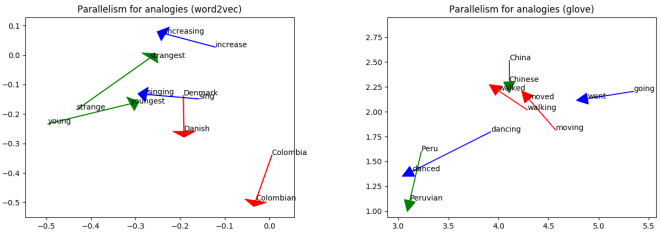
3.2 Analogies protocol

In this subsection we show that the protocol we described in Sect. 3.1 for finding analogies does not really work in general. We ran it on 50 word triplets (a, b, c) as input, with $k = 10$ in the k-NN stage, but only obtained 35 correct valid analogies, namely those in Fig. 2.

3.3 Turning the protocol into an algorithm

The protocol described in Sect. 3.2 is termed "protocol" rather than "algorithm" because it involves a human interaction when choosing the appropriate word out of the set of $k = 5$ nearest neighbours to $v_c + (v_b - v_a)$. Since natural language processing tasks usually concern sets of words of higher cardinalities than humans can handle, we are interested in an algorithm for finding analogies rather than a protocol. In this section we present an algorithm which takes the human decision out of the protocol sketched above. Then we show that this algorithm has the same shortcomings as the protocol, as shown in Sect. 3.2.

We first remark that the obvious way to turn the protocol of Sect. 3.2 into an algorithm is to set $k = 1$ in the k-NN stage, which obviously removes the need for a human choice. If we do this, however, we cannot even complete the famous "king:man=queen:woman" analogy: instead of "woman", we actually get "king" using glove embeddings.



(a) word2vec (b) glove

(c) fastText

Fig. 1. Sanity check

'Athens:Greece=Baghdad:Iraq', 'Ottawa:Canada=Islamabad:Pakistan'
'Ashgabat:Turkmenistan=Athens:Greece', 'Beirut:Lebanon=Bern:Switzerland',
'Bujumbura:Burundi=Conakry:Guinea', 'Doha:Qatar=Hanoi:Vietnam',
'his:her=brothers:sisters', 'easy:easier=simple:simpler',
'low:lower=tight:tighter', 'strong:stronger=bad:worse',
'cold:coldest=low:lowest', 'discover:discovering=enhance:enhancing',
'play:playing=sing:singing', 'think:thinking=implement:implementing',
'Cambodia:Cambodian=Croatia:Croatian', 'Greece:Greek=Italy:Italian',
'Mexico:Mexican=Portugal:Portuguese', 'Sweden:Swedish=Austria:Austrian',
'flying:flew=jumping:jumped', 'looking:looked=screaming:screamed',
'selling:sold=taking:took', 'thinking:thought=flying:flew',
'child:children=snake:snakes', 'mouse:mice=computer:computers',
'search:searches=work:works'

Fig. 2. Some valid analogies following Protocol 3.2

Following our first definition in Eq. (1), we instead propose the notion of parallelism in Eq. (9):

$$||v_d - v_c - (v_b - v_a)|| \leq \tau \max(||v_b - v_a||, ||v_d - v_c||) \tag{9}$$

where τ is a small scalar. Eq. (9) is a sufficient condition for quasi-parallelism between $v_d - v_c$ and $v_b - v_a$. The algorithm is very simple: given quadruplets (a, b, c, d) of words, and tag the quadruplet as a valid analogy if Eq. (9) is satisfied. We also generalize the PCA dimensional reduction from 2D to more dimensionalities.

We ran this algorithm on a database of quadruplets corresponding to valid analogies, and obtained the results in table 1. The fact that the results are surprisingly low was one of our initial motivations for this work. The failure of this algorithm indicates that parallelism in analogies may be more incidental than systematic.

Dimension	word2vec		glove		fastText	
	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.1$	$\tau = 0.2$
2	1.08 %	5.17 %	3.34 %	12.93 %	0.97 %	4.92 %
10	0.00 %	0.00%	0.00 %	0.09 %	0.00 %	0.00%
20	0.00 %	0.00 %	0.00 %	0.00%	0.00 %	0.00%
50	0.00 %	0.00%	0.00 %	0.00%	0.00 %	0.00%
100	0.00 %	0.00 %	0.00 %	0.00%	0.00 %	0.00%
300	0.00 %	0.00 %	0.00 %	0.00%	0.00 %	0.00%

Table 1. Analogies from parallelism, F1-score

3.4 Supervised classification:

The failure of an algorithm for correctly labelling analogies based on parallelism of difference vectors (see Sect. 3.3) does not necessarily imply that analogies correctly labeled (at least approximately) using other means. In this section we propose a very common supervised learning approach (a simple $k-$NN).

More precisely, we trained a $5-$NN to predict analogies using vector differences, following Eq. (1). If (a, b, c, d) is an analogy quadruplet, we use the representation:

$$x_{abcd} = (v_b - v_a, v_d - v_c) \tag{10}$$

to predict the class of the quadruplet (a, b, c, d) (either no relation or being the capital of, plural, etc). If the angles between the vectors $v_b - v_a$ and $v_d - v_c$ (hint of parallelism) contain important information with respect to analogies, this representation should yield a good classification score. The dataset used is composed of 13 types of analogies, with thousand of examples in total.[2]

[2] Link to repository

We considered 1000 pairs of words sharing a relation, with 13 labels (1 to 13, respectively: capital-common-countries and capital-world (merged), currency, city-in-state, family, adjective-to-adverb, opposite, comparative, superlative, present-participle, nationality-adjective, past-tense, plural, plural-verbs), and 1000 pairs of words sharing no relation (label 0). In order to generate different random quadruplets, we ran 500 simulations. Average results are in Table 2.

Dimension	word2vec	glove	fastText
2	62.47 %	69.30 %	68.74 %
10	86.44 %	85.62 %	90.40 %
20	74.74 %	77.45 %	80.57 %
50	55.11 %	61.24 %	55.30 %
100	50.57 %	51.26 %	50.56 %
300	51.12 %	51.72 %	49.98 %

Table 2. Multi-class F1 score classification of analogies based on representation 10 (5-nearest neighbors)

The results in Table 2 suggest that the representations obtained from Eq. (10) allow a good classification of analogies in dimension 10 when Euclidean geometry is used with a $5-$NN. However, in the remaining dimensions, vector differences does not encode enough information with regards to analogies.

4 Parallelism for analogies with graph propagation

In this section we present an algorithm which takes an existing word embedding as input, and outputs a modified word embedding for which analogies correspond to a notion of parallelism in vector differences. These new word embeddings will be later used (see Sect. 5) to contradict the hypothesis that analogies corresponding to parallel vector differences does not make the word embedding better for common classification tasks.

Let us consider a family of semantic relations $(\mathcal{R}_k | 1 \leq k \leq r)$. For instance, this family can contain the plural or superlative relation. One of the relations $\mathcal{R}_k$ creates the analogy $a : b = c : d$, if and only if: $a\mathcal{R}_k b$ and $c\mathcal{R}_k d$, i.e semantic relations create quadruplets of analogies in the following sense:

$$(a, b, c, d) \text{ is an analogy quadruplet} \iff \exists k,\ a\mathcal{R}_k b \text{ and } d\mathcal{R}_k c \tag{11}$$

A sufficient condition for relation (1) to hold for a quadruplet is for each pair a, b in the relation $\mathcal{R}_k$:

$$\exists \mu_k \in \mathbb{R}^d\ ,\quad a\mathcal{R}_k b \iff v_b = v_a + \mu_k \tag{12}$$

Eq. (12) can be generalized to other functions than summing a constant vector, namely it suffices that

$$\exists f_k : \mathbb{R}^d \longrightarrow \mathbb{R}^d \ , \quad v_a \mathcal{R}_k v_b \iff v_b = f_k(v_a) \tag{13}$$

Other choices of f_k might be interesting, but are not considered in this work.

In order to generate word vectors satisfying Eq. (12), we propose a routine using propagation on graphs. The first step consists in building a directed graph of words (V, E) encoding analogies:

$$(i, j) \in E \Leftrightarrow \exists k \ (i \mathcal{R}_k j) \tag{14}$$

We suppose that the relations we consider induce at most one relation between two given words. This corresponds to intuition and is verified with the relations we consider in our experiments (however, it should be discussed for any family of relations). Therefore, we can label each edge with the type k of analogy involved (namely being the capital of, plural, etc, ...). Then, we use a graph propagation algorithm (algorithm 1) involving Eq. (12) relation. We remark that propagation requires initial node representations.

Algorithm 1: Graph propagation for analogies

Data: List of relations, vectors $\mu_1, ..., \mu_K \in \mathbb{R}^d$
Result: New representations
1 Build graph G of analogies (Eq. (14));
2 Extract connected components $C_1, ..., C_c$ from G;
3 **for** $j = 1 \to c$ **do**
4 | Select source node $s_1 \in C_j$;
5 | $v_{s_1} \leftarrow$ Generate initial representation of s_1;
6 | $s_2, ..., s_{|C_j|} \leftarrow$ Breadth first search from s_1;
7 | **for** $r = 2 \to |C_j|$ **do**
8 | | $k \leftarrow$ index of relation between s_r and s_{r+1};
9 | | $v_{s_{r+1}} = v_{s_r} + \mu_k$;
10 | **end**
11 **end**
12 Return $(v_i \mid 1 \leq i \leq |G|)$

Proposition 1. *Let G the graph of analogies. If G is a forest, then the representations obtained with Algorithm 1 verify Eq.* (12)*.*

Proof. A forest structure implies the existence of a source node s for each component in G. For each component, every visited node with breadth-first search starting from s has only one parent, so the update defined Line 9 in Algorithm 1 defines a representation that verify Eq. (12) for the current node and its parent. □

However, if G is not a forest, words can have several parents. In this case, if $(parent_1, child)$ is visited before $(parent_2, child)$, our graph propagation method will not respect Eq. (12) for $(parent_1, child)$. This is the case with homonyms. For example, Peso is the currency for Argentina, but the currency for Mexico too. In practice, we selected $\mu_1, ..., \mu_K$ as a family of independent vectors in $\mathbb{R}^d$. We found better results in our experiments with $\forall i,\ ||\mu_i|| \geq d$. This can be explained by the fact that the vectors of relations needs to be non negligible when compared to difference of the words vectors.

5 Experiments with new embeddings

In this section we present results of the experiments described in Sec. 3 with the updated embeddings obtained with the propagation Algorithm 1. We call $X++$ the new word embeddings obtained with the propagation algorithm from the word embeddings X.

5.1 Classification of analogies

Analogies from parallelism: As in section 3.3 using Eq. (9). Results are in table 3. F1-scores are almost perfect (by design) in all dimensions.

Dimension	word2vec++		glove++		fastText++	
	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.1$	$\tau = 0.2$
2	96.80 %	96.50 %	97.92 %	97.15 %	98.15 %	97.61 %
10	98.48%	98.54 %	97.88 %	97.88 %	98.25 %	98.31 %
20	98.12 %	98.18 %	98.14 %	98.43 %	96.56 %	96.56%
50	96.80 %	96.80%	98.28 %	98.36 %	98.17 %	98.17%
100	98.08 %	98.08 %	98.19 %	98.19 %	98.06 %	98.06 %
300	98.41 %	98.41 %	98.40 %	98.40 %	98.30 %	98.30 %

Table 3. Analogies from parallelism with updated embeddings, F1-score

With supervised learning: Same experiments as in Sec. 3.4: 1000 pairs of words sharing a relation with 13 labels (1 to 13), and 1000 pairs of words sharing no relation (label 0). Results are in table 4.

5.2 Text classification: comparison using KNN

We used three datasets: one for binary classification (Subjectivity) and two for multi-class classification (WebKB and Amazon)). For reasons of time computation we used a subset of WebKB and Amazon datasets (500 samples). The implementation and datasets are available online[3]. Results are in table 5.

[3] Link to repository

Dimension	word2vec++	glove++	fastText++
2	99.73 %	99.44 %	99.31 %
10	99.75 %	99.36 %	99.64 %
20	99.80 %	99.52 %	99.94 %
50	99.56 %	99.63 %	99.49 %
100	99.89 %	99.54 %	99.42 %
300	99.40 %	99.86 %	99.45 %

Table 4. Multi-class F1 score on classification of analogies based on relation 10 with updated embeddings (5-nearest neighbors)

	word2vec	glove	fastText
Subjectivity	81.69 %	81.02 %	82.14 %
WebKB	71.50 %	71.00 %	70.50 %
Amazon	65.20 %	63.60 %	60 %
	word2vec++	glove++	fastText++
Subjectivity	81.69 %	80.38 %	81.57 %
WebKB	71.50 %	72.00 %	72.00 %
Amazon	65.20 %	61.00 %	56.40 %

Table 5. Text classification ($d = 20$), F1-score

6 Conclusion

In this paper we discussed the well-advertised "geometrical property" of word embeddings w.r.t. analogies. By using a corpus of analogies, we showed that this property does not hold in general, in two or more dimensions. We conclude that the appearance of this geometrical property might be incidental rather than systematic or even likely.

This is somewhat in contrast to the theoretical findings of [1]. One possible way to reconcile these two views is that the concentration of measure argument in [1, Lemma 2.1] might yield high errors in vectors spaces having dimension as low as $\mathbb{R}^{300}$. Using very high-dimensional vector spaces might conceivably increase the occurrence of almost parallel differences for analogies. By the phenomenon of *distance resolution*, however, algorithms based on finding closest vectors in high dimensions require computations with ever higher precision when the vectors are generated randomly. Moreover, the model of [1] only warrants approximate parallelism. So, even if high dimensional word vectors pairs were almost parallel with high probability, verifying this property might require considerable computational work related to floating point precision.

By creating word embeddings on which the geometrical property is enforced by design, we also showed empirically that the property appears to be irrelevant w.r.t. the performance of a common information retrieval algorithm (k-NN). So, whether it holds or not, unless one is trying to find analogies by using the

property, is probably a moot point. We obviously grateful to this property for the (considerable, but unscientific) benefit of having attracted some attention of the general public to an important aspect of computational linguistics.

References

1. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: A latent variable model approach to pmi-based word embeddings. Transactions of the Association for Computational Linguistics **4**, 385–399 (2016)
2. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: Linear algebraic structure of word senses, with applications to polysemy. Transactions of the Association of Computational Linguistics **6**, 483–495 (2018)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Advances in Neural Information Processing Systems. pp. 4349–4357 (2016)
5. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
6. Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. Science **356**(6334), 183–186 (2017)
7. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational linguistics **16**(1), 22–29 (1990)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Hatcher, E., Gospodnetic, O.: Lucene in action. Manning Publications (2004)
10. Jastrzebski, S., Leśniak, D., Czarnecki, W.M.: How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. arXiv preprint arXiv:1702.02170 (2017)
11. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. vol. 15, pp. 2181–2187 (2015)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
13. Mikolov, T., Yih, W.t., Zweig, G.: 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. pp. 746–751 (2013)
14. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
15. Nissim, M., van Noord, R., van der Goot, R.: Fair is better than sensational: Man is to doctor as woman is to doctor. arXiv preprint arXiv:1905.09866 (2019)
16. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
17. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)

18. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. vol. 242, pp. 133–142. Piscataway, NJ (2003)
19. Salton, G.: The smart system. Retrieval Results and Future Plans (1971)
20. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18**(11), 613–620 (1975)
21. Vylomova, E., Rimell, L., Cohn, T., Baldwin, T.: Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. arXiv preprint arXiv:1509.01692 (2015)
22. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. vol. 14, pp. 1112–1119 (2014)