Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4789–4797
Marseille, 11–16 May 2020

On the Correlation of Word Embedding Evaluation Metrics

François Torregrossa, Vincent Claveau, Nihel Kooli,
Guillaume Gravier, Robin Allesiardo
Solocal-IRISA, IRISA-CNRS, Solocal,
IRISA-CNRS, Solocal
Rennes
ftorregrossa@solocal.com, vincent.claveau@irisa.fr, nkooli@solocal.com,
guillaume.gravier@irisa.fr, rallesiardo@solocal.com

Abstract

Word embeddings intervene in a wide range of natural language processing tasks. These geometrical representations are easy to manipulate for automatic systems. Therefore, they quickly invaded all areas of language processing. While they surpass all predecessors, it is still not straightforward why and how they do so. In this article, we propose to investigate all kind of evaluation metrics on various datasets in order to discover how they correlate with each other. Those correlations lead to 1) a fast solution to select the best word embeddings among many others, 2) a new criterion that may improve the current state of static Euclidean word embeddings, and 3) a way to create a set of complementary datasets, i.e. each dataset quantifies a different aspect of word embeddings.

1. Introduction

Word embeddings are continuous vector representations of word paradigmatics and syntagmatics. Since they capture multiple high-level characteristics of language, their evaluation is particularly difficult: it usually consists of quantifying their performance on various tasks. This process is thorny because the outcome value does not explain entirely the complexity of these models. In other words, a model performing well under a specific evaluation might poorly work for a different one (Schnabel et al., 2015). As an example, some word embedding evaluations promote comparison of embeddings with human judgement while others favour embeddings behaviour analysis on downstream tasks, as pointed by Schnabel et al. (2015).

In this work, we propose to investigate correlations between numerous evaluations for word embedding. We restrict the study to FastText embeddings introduced by Bojanowski et al. (2017), but this methodology can be applied to other kinds of word embedding techniques. The understanding of evaluation correlations may provide several useful tools:

- Strongly correlated evaluations raise a question on the relevance of performing these. Actually, it could be possible to only keep one evaluation among the correlated evaluation set, since its score would directly affect the score of others. Therefore, it could reduce the number of needed evaluations.
- Inexpensive evaluation processes correlated with time-consuming ones could be helpful to speed up optimisation of hyper-parameters. Indeed, they could be used to bypass those demanding steps, thus, saving time.
- Some evaluations do not require any external data since they look into global structure of vectors as presented in Tifrea et al. (2018) and Houle et al. (2012). If related to other tasks, these data-free metrics could be incorporated into the optimisation process in order to improve the performance on related tasks.

The article is organised as follows. Section 2 compares our proposed methodology to the current state of the art of word embeddings evaluation. Section 3 introduces the evaluation processes and materials we used for this investigation. Then section 4 details the experimental setup and discusses the results of experiments. The final section presents some conclusive remarks about this work.

2. Related Work

Evaluations of word embeddings is not a new topic. Many resources and procedures, some used in this work and others exhaustively listed by Bakarov (2018), have been proposed in order to compare various methods such as GloVe (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013a). Quickly, the distinction between intrinsic and extrinsic evaluations was made, as stated by (Schnabel et al., 2015). The first one being related to the word embedding itself whereas the second uses it as an input of another model for a downstream task.

Generally extrinsic evaluations are more crucial than intrinsic ones. Actually, extrinsic evaluations often are the ultimate goal of language processing while intrinsic evaluations try to estimate the global quality of language representations. Some work (Schnabel et al., 2015) unsuccessfully try to identify correlations between extrinsic and intrinsic scores, using word embeddings computed with different methods. However, intrinsic and extrinsic scores from word embeddings calculated with the same method, as done by Qiu et al. (2018), are significantly correlated. We propose to prolong their work to English word embeddings and to more popular datasets. In fact, comparing embeddings from different classes is thorny since different algorithms catch different language aspects. As shown by Claveau and Kijak (2016), some embeddings could be created in order to solve specific tasks while neglecting other language aspects. This is why, we only investigate word embeddings learned using a unique algorithm: FastText (Bojanowski et al., 2017).

Another aspect treated in this work is the introduction of global metrics which are metrics trying to catch intrinsic structures in vectors, with no data other than these vectors. Tsvetkov et al. (2015) proposed a metric trying to automatically understand the meaning of vector dimensions. Their metric show good correlation with both intrinsic and extrinsic evaluations, but still requires data. We propose to extend this work by taking data-free matrix analysis techniques from signal processing and computer vision (Poling and Lerman, 2013; Roy and Vetterli, 2007). The major interest in data-free metrics is that they can be introduced during the learning phase as a regularisation term.

3. Evaluation Metrics

In this section, we present three categories of metrics used to evaluate embeddings: global, intrinsic and extrinsic. For each category, we highlight the datasets used for the experiments. We denote by $\mathcal{E}$ the embedding and $W \in \mathbb{R}^{N \times D}$, the word embedding matrix of $\mathcal{E}$, where N is the number of words in the vocabulary and D is the dimension. Consequently, the i-th row of W is a vector with D features representing the i-th word of the vocabulary.

3.1. Global Metrics

Global metrics are data-free (i.e., with no external data other than W) evaluations finding out relationships between vectors or studying their distribution. We propose to see two category here.

3.1.1. Global Intrinsic Dimensionality

Intrinsic Dimensionality (ID) is a local metric, used in information retrieval and introduced by Houle et al. (2012), aiming to be related to the *complexity* of the neighbourhood of a query point x. This *complexity* being the minimal dimensionality required to describe the data points falling within the intersection I of two concentric spheres of centre x. As highlighted by Claveau (2018), high dimensionality indicates that I structure is complex and therefore means that a slight shift on x would completely change the nearest neighbours, leading to poor accuracy in search tasks (as analogy, see Section 3.2.). In other words, neighbours of x and $x + \epsilon$ are totally different (where ϵ is a noise vector, with $||\epsilon|| \ll 1$).

An estimated value of local ID of x can be computed on $\mathcal{E}$ using the maximum likelihood estimate following Amsaleg et al. (2015). Thus, noting $\mathcal{N}_{||.||_2}(x, k)$ the k nearest neighbours of x in $\mathcal{E}$ (using the L2-norm), its formulation is:

$$\mathrm{ID}_x = - \left[\frac{1}{k} \cdot \sum_{y \in \mathcal{N}_{||.||_2}(x,k)} \ln \frac{||y - x||_2}{1 + \max\limits_{z \in \mathcal{N}_{||.||_2}(x,k)} ||z - x||_2} \right]^{-1} .$$

This estimate is local and only describes the complexity of the surroundings of a word vector x. We propose to create global metrics by studying the distribution of $(\mathrm{ID}_x)_{x \in \mathcal{E}}$, as done by Amsaleg et al. (2015). For instance, the mean, median, standard deviation or percentiles of this distribution. Our intuition is that embeddings containing a large number of query points with simple neighbourhoods are likely to perform well on analogy and semantic tasks. On the contrary, widespread complex neighbourhoods would plummet the accuracy.

A similar approach to the ID based on distance is the ID based on similarity. Instead of the L2-norm, the dot product, often employed for word embeddings, can be used as follows:

$$\mathrm{ID}_x = - \left[\frac{1}{k} \cdot \sum_{y \in \mathcal{N}_{\langle .,. \rangle}(x,k)} \ln \frac{\langle x, y \rangle}{1 + \max\limits_{z \in \mathcal{N}_{\langle .,. \rangle}(x,k)} \langle x, z \rangle} \right]^{-1} .$$

In the following, we name those set of metrics: *Global Intrinsic Dimensionality Metrics (GIDM)*.

3.1.2. Effective rank and Empirical dimension

Metrics from computer vision and signal processing can be used to quantify the number of significative dimensions of the word embedding W. This quantity can be expressed by singular values since they indicate the principal axis of variance of vectors composing W. It can be formulated in many ways. First Roy and Vetterli (2007) proposed the *effective rank* (erank):

$$\mathrm{erank}(W) = \exp \left(- \sum_{i=1}^{D} \left[\frac{s_i}{\sum_{j=1}^{D} s_j} \cdot \log \left(\frac{s_i}{\sum_{j=1}^{D} s_j} \right) \right] \right) ,$$

where $\boldsymbol{s} = (s_i)_{i \in \, [\![1, D]\!]}$ are the singular values of W.

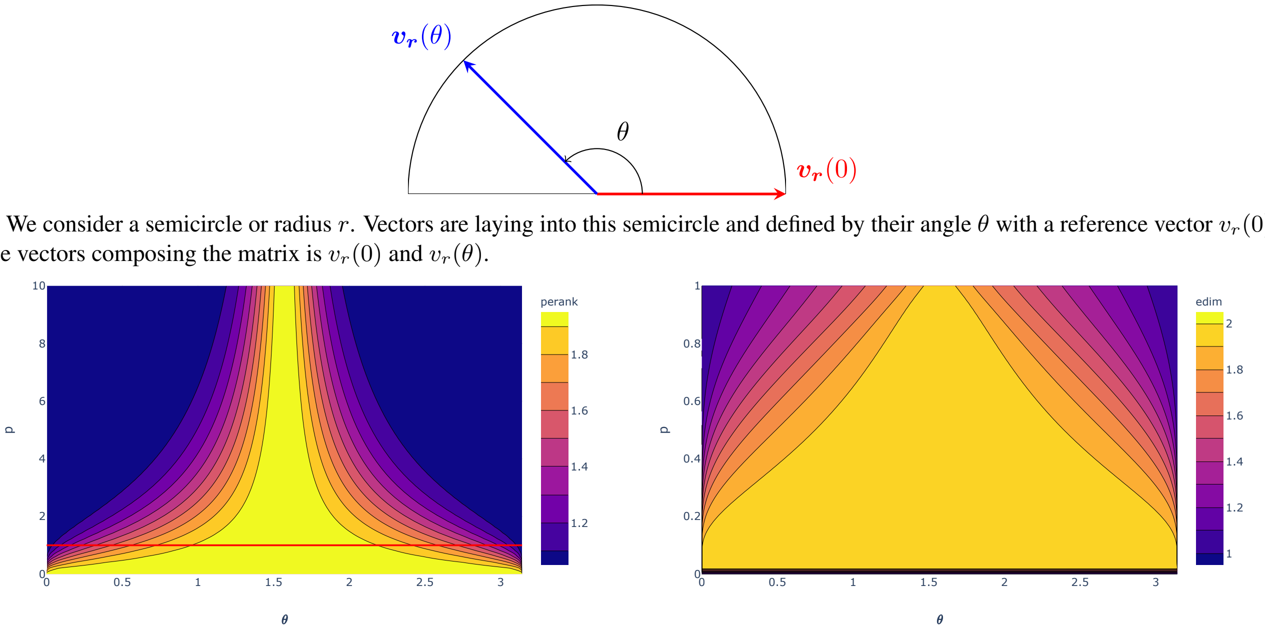
One can notice that the effective rank uses the Shannon entropy of singular values to measure the quantity of information held by each of them. Ideally, singular values should carry similar amount of information (high entropy) since a preponderant singular value (low entropy) indicates poor usage of the dimensionality of the embedding space. In fact, low entropy points out that vectors can be encoded into a unidimensional space since vectors of W are well scattered on the axis attached to this preponderant singular value. Hence it highlights under-training, as low entropy indicates low information encoding.

This metric is convenient since $\forall M \in \mathcal{R}^{N \times D}, \mathrm{erank}(M) \in [1, D]$. A value close to 1 corresponds to low entropy thus the matrix can be compressed into a unidimensional space, while a value close to D indicates the opposite. Values in-between are closely equal to the minimum number of dimensions needed to compress the vectors of W with a low reconstruction error, as shown by Roy and Vetterli (2007).

However, for some use cases, the effective rank tends to overestimate this minimum number of dimensions. This is why Poling and Lerman (2013) proposed the *empirical dimension* (edim), introducing a variable parameter $p \in [0, 1]$. This parameter aims to control the estimation and is expressed as follows:

$$\mathrm{edim}(W, p) = \frac{||\boldsymbol{s}||_p}{||\boldsymbol{s}||_{\frac{p}{1-p}}},$$

where $||\boldsymbol{x}||_p = (\sum_i x_i^p)^{\frac{1}{p}}$.



(a) We consider a semicircle or radius r. Vectors are laying into this semicircle and defined by their angle θ with a reference vector $v_r(0)$. The vectors composing the matrix is $v_r(0)$ and $v_r(\theta)$.

(b) Powered effective rank (perank), $p \in [0, 10]$. The horizontal red line corresponds to erank (special case $p = 1$).

(c) Empirical dimension (edim), $p \in [0, 1]$.

Figure 1: Toy example to inspect perank and edim. A matrix containing two vectors of two dimensions is created taking the vectors of 1a. Values of perank and edim are computed while θ varies in $[0, \pi]$.

The case $p = 1$ shows strong correlations between edim and intrinsic and extrinsic tasks. However it is not possible to go beyond $p = 1$ with the empirical dimension, despite the fact that the function seems expandable over this value. To inspect this domain, we propose another estimator, the *powered effective rank* (perank):

$$\mathrm{perank}(W, p) = \exp\left(-\sum_{i=1}^{D}\left[\frac{s_i^p}{\sum_{j=1}^{D} s_j^p}\cdot\log\left(\frac{s_i^p}{\sum_{j=1}^{D} s_j^p}\right)\right]\right),$$

where p varies continuously in $[-\infty, +\infty]$.

Another interpretation of these metrics is as a criterion of orthogonality. Indeed their maximum is reached for orthogonal matrices. As shown in Figure 1, $\theta = \frac{\pi}{2}$ is the argmax value of these functions, while p seems to control the sensitivity of the metric to orthogonality. In fact it is possible to show the following:

$$\forall p \neq 0,\ \ \mathrm{perank}(W, p) = D \vee \mathrm{edim}(W, p) = D$$
$$\Leftrightarrow \qquad \frac{1}{\max\limits_{i \in [\![1, D]\!]} s_i} \cdot W \text{ is semi-orthogonal.}$$

This is a useful result since orthogonality regularisation can be introduced during the optimisation process as in Bansal et al. (2018), Arora et al. (2019) and Zhang et al. (2018). Therefore, if these metrics are correlated to good performance of word embeddings, compel orthogonality would help learning effective models.

3.2. Intrinsic Evaluations

Intrinsic evaluations compare embedding structures to human judgement. They need external data to carry out this comparison and mainly assess simple language concepts. In this work, we study three different kinds of intrinsic evaluations focusing on different language aspects. The cosine similarity:

$$\cos(v_A, v_B) = \frac{\langle\, v_A, v_B \,\rangle}{||v_A||\,||v_B||} \tag{1}$$

is the metric used to compare two vectors v_A and v_B from the word embedding W.

Below we discuss three common word embedding evaluation methods, namely : similarity, analogy and categorisation.

3.2.1. Similarity

Similarity consists of scoring word pairs. Each pair is human-labelled with a score representing the compatibility between the concepts of the pair. This compatibility score is specific to datasets and often characterises the synonymy (Finkelstein et al., 2002; Hill et al., 2015; Gerz et al., 2016; Rubenstein and Goodenough, 1965) or the entailment (Vulić et al., 2017).

The evaluation relies on measuring the Spearman correlation between labelled scores and reconstructed scores from the word embedding. The reconstructed scores are obtained by taking the cosine similarity (1) between pairs. The correlation score constitutes in the end the value of the evaluation. Similarity datasets used in this study are reported on Table 1. The majority of them are datasets using synonymy (i.e. semantic proximity) as a guide to estimate the score. We add HyperLex from Vulić et al. (2017) to introduce another aspect of language in the evaluation process: entailment. For instance the $type\_of$ or is_a relation: a *duck* is an *animal*, but the opposite is not always true.

Name	Size	Pairwise Score based on:
WordSim353 (Finkelstein et al., 2002)	353	Synonymy (common words)
MEN (Bruni et al., 2014)	3000	Synonymy (common words)
RG (Rubenstein and Goodenough, 1965)	65	Synonymy (common words)
SimLex-999 (Hill et al., 2015)	999	Synonymy (common words)
SimVerb (Gerz et al., 2016)	3500	Synonymy (essentially verbs)
RareWords (RW) (Luong et al., 2013)	2034	Synonymy (low frequency words)
HyperLex (Vulić et al., 2017)	2616	Entailment (common words)

Table 1: Similarity datasets used in this work (Bakarov, 2018).

Name	Size	Relation Types
Google Analogy (Mikolov et al., 2013a)	19000	Capital, Country, Family, Currency, Cities, Morphology
MSR (Mikolov et al., 2013c)	8000	Morphology

Table 2: Analogy datasets used in this work (Bakarov, 2018).

3.2.2. Analogy

Analogy, proposed by Mikolov et al. (2013b), assesses the embedding of any kind of relationships. Given three words w_A, w_B and w_C, such that w_A is related to w_B through a relation $\mathcal{R}$, the task consists of finding a fourth word, w_D, that is related to w_C through the same relation $\mathcal{R}$. Technically, w_D is found as a solution of the problems formulated by Levy et al. (2015), leveraging the cosine similarity (1). We consider the two analogy datasets detailed in Table 2.

3.2.3. Categorisation

Categorisation is a reconstruction exercise aiming to recover semantic clusters in the embedding space. The dataset is composed of K clusters and M words. The goal is to reconstruct K clusters using the M word vectors of the embedding. The reconstruction can be done with any clustering algorithm, but Schnabel et al. (2015) suggests using CLUTO toolkit from Karypis (2003) with default parameters. In this setting, CLUTO algorithm iteratively decomposes word vectors in K clusters and maximises the cosine similarity of words from the same cluster. After the clustering step, we compute the difference between the ground-truth clusters with the reconstructed ones. This is achieved with the purity metric. For the evaluation of the embedding on this task, we use three datasets (see Table 3).

Name	Size	Number of clusters
Battig (Baroni et al., 2010)	5330	56
AP (Almuhareb, 2006)	402	21
BLESS (Baroni and Lenci, 2011)	200	17

Table 3: Categorisation datasets for intrinsic evaluations (Bakarov, 2018).

3.3. Extrinsic Evaluations

Extrinsic evaluations are the last sort of evaluations considered in this work. They focus on more complex language aspects. Hence, they need external data and an additional language modelling step. Among the long list of extrinsic tasks, we chose three of them that cover a large spectrum of language skills, namely: Named Entity Recognition (NER), Sentiment Analysis (SA) and Text Classification (TC). Only one dataset for each task is used since extrinsic evaluations are particularly time and resource demanding.

3.3.1. Named Entity Recognition

Named entity recognition (Li et al., 2018) investigates the capacity of models to extract high-level information from plain text data. It asks models to recover entity class from entity mention in text. In the end, each word has to be classified in different categories representing entities.

CoNLL2003 is the NER dataset considered for this work. We followed Sang and Meulder (2003) guidelines to set up the experiment. This dataset is made of sentences extracted from news thread. Words are then labelled by 5 sort of entities: O (None), PER (person), ORG (organisation), LOC (location), MISC (miscellaneous). Training and development sets are used for training while test set is kept for evaluation.

3.3.2. Sentiment Analysis

In our case, sentiment analysis is a sentence-level classification problem for an opinion text (Joshi et al., 2017). Sentences have to be classified as positive, negative or neutral.

The Stanford Sentiment Treebank dataset (SST), proposed by Socher et al. (2013), is chosen for this task. Each sentence is a movie review labelled by its global judgement: very positive, positive, neutral, negative or very negative. The objective is to recover sentiment classes from sentences, and measure the average accuracy. This setup is known as SST-1 (Zhou et al., 2016). Here again, train and dev splits are involved for training whereas test split is kept for evaluation.

3.3.3. Text Classification

This task is similar to sentiment analysis, since models have to classify documents into different categories (Kowsari et al., 2019). The difference relies in the meaning of the labels and the nature of the text. They characterise high-level topics.

The AGNews dataset is a data source found online[1] and used

[1] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

Name	Distribution (uniform choice)
Dimension (-dim)	$[20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]$
Learning Rate (-lr)	$[5 \cdot 10^{-2}, 5 \cdot 10^{-3}, 5 \cdot 10^{-4}]$
Windows Size (-ws)	$[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]$
Number of Epoch (-epoch)	$[1, 2, 3, 4, 5]$
Negative Sampling Size (-neg)	$[2, 4, 6, 8, 10, 12, 14]$
Minimum Frequency (-minCount)	$[5, 50, 100, 250, 500]$
Number of Bucket (-bucket)	$[1, 5 \cdot 10^3, 10^4, 5 \cdot 10^4, 10^5, 5 \cdot 10^5, 10^6, 2 \cdot 10^6]$
Min N-gram (-minn)	$[2, 3, 4]$
Max N-gram (-maxn)	$\mathrm{minn} + [0, 1, 2, 3]$
Subsampling Threshold (-t)	$[5 \cdot 10^{-1}, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$
Training Corpus (-input)	10%, 25% of random Wikipedia articles or whole.

Table 4: Distributions of hyper-parameters used to generate FastText embeddings.

for our experiments. It is composed of news articles falling in 4 categories: World, Sports, Business and Sci/Tech.

So far we defined every kind of evaluations we need (global, intrinsic and extrinsic) as well as datasets we use in the following. The aim of our experiment is to highlight correlation between those kinds of metrics.

4. Experiments

This section details the setup of our experiments. It provides an overview of the way we produced word embeddings, how we handled additional modelling for extrinsic tasks and finally presents the results.

4.1. Embeddings Generation Process

FastText (Bojanowski et al., 2017) is a method extending Word2Vec (Mikolov et al., 2013a) using morphology of words to compute their vector representations. We chose the SkipGram version of FastText to produce word embeddings because this method is very generic, hence, usable as a good proxy for other embedding methods. Particularly, it includes Word2Vec.

Using the FastText Library[2], we generated 140 FastText embeddings with different sets of hyper-parameters. Those are randomly sampled from distributions listed in Table 4. With this method, we created various FastText embeddings with different handicaps and assets. For instance, the window size influences the ability to comprehend paradigmatic and syntagmatic aspects. The training corpus is also a hyper-parameter and is based on Wikipedia dumps[3] being either complete or downsampled (around 10% or 25% of the original size).

At the end of the training phase, we extracted the first 200,000 most frequent words from the FastText embeddings in order to compute global metrics and perform the analogy task. For extrinsic tasks, word representations are calculated with the FastText models.

4.2. Additional Modelling for extrinsic tasks

As mentioned above, extrinsic evaluations cannot be carried with the word embedding alone. It requires an additional model (as neural networks) which incorporates external data from the training corpus of the task and extract task-proprietary knowledge from input word vectors. For each extrinsic task, we fixed an architecture and its parameters. Therefore, the only variable considered is the input word embeddings. Models are implemented with the Pytorch framework (Paszke et al., 2017) and remain relatively simple since we are not interested in state of the art performance but in variations of the performance with regard to input word embeddings.

Named entity recognition. The BiLSTM-CRF architecture, proposed by Lample et al. (2016), is chosen for this task. We replaced LSTM by GRU for simplicity without compromising the performance of the architecture as shown by Chung et al. (2014). The CRF layer is taken from AllenNLP (Gardner et al., 2017). We fixed the number of BiGRU layers to 1 with 2x256 units. Before being fed to the CRF, a linear layer turns the 512 features of the BiGRU to 5 features corresponding to the classes of the dataset. This model achieves near state-of-the-art performances (91.27% F1score) with a 300-dimensional FastText trained following Bojanowski et al. (2017).

Sentiment analysis. A BiGRU with identical hyper-parameters (as NER) is chosen for this evaluation. The last hidden vectors of both directions are concatenated, such that the input sentence is turned into a vector. A linear layer and a softmax transform this vector into a vector of probabilities indexed by sentiment classes.

Text classification. Sentiment analysis model is used here. All text is passed into the BiGRU model and the last hidden layer is used to infer text classes. Instead of sentiment classes, the output is a vector of probabilities on topic classes.

4.3. Results

Each FastText embedding is assessed by evaluations mentioned in Section 2. Therefore, each word embedding is represented by the output scores of evaluation procedures.

[2] `https://github.com/facebookresearch/fastText`

[3] `https://dumps.wikimedia.org/wikidatawiki/`

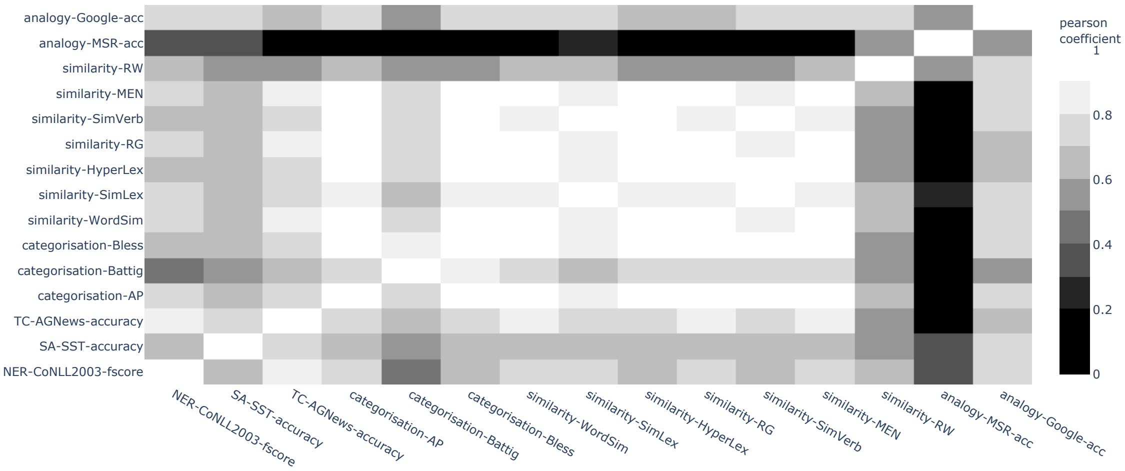
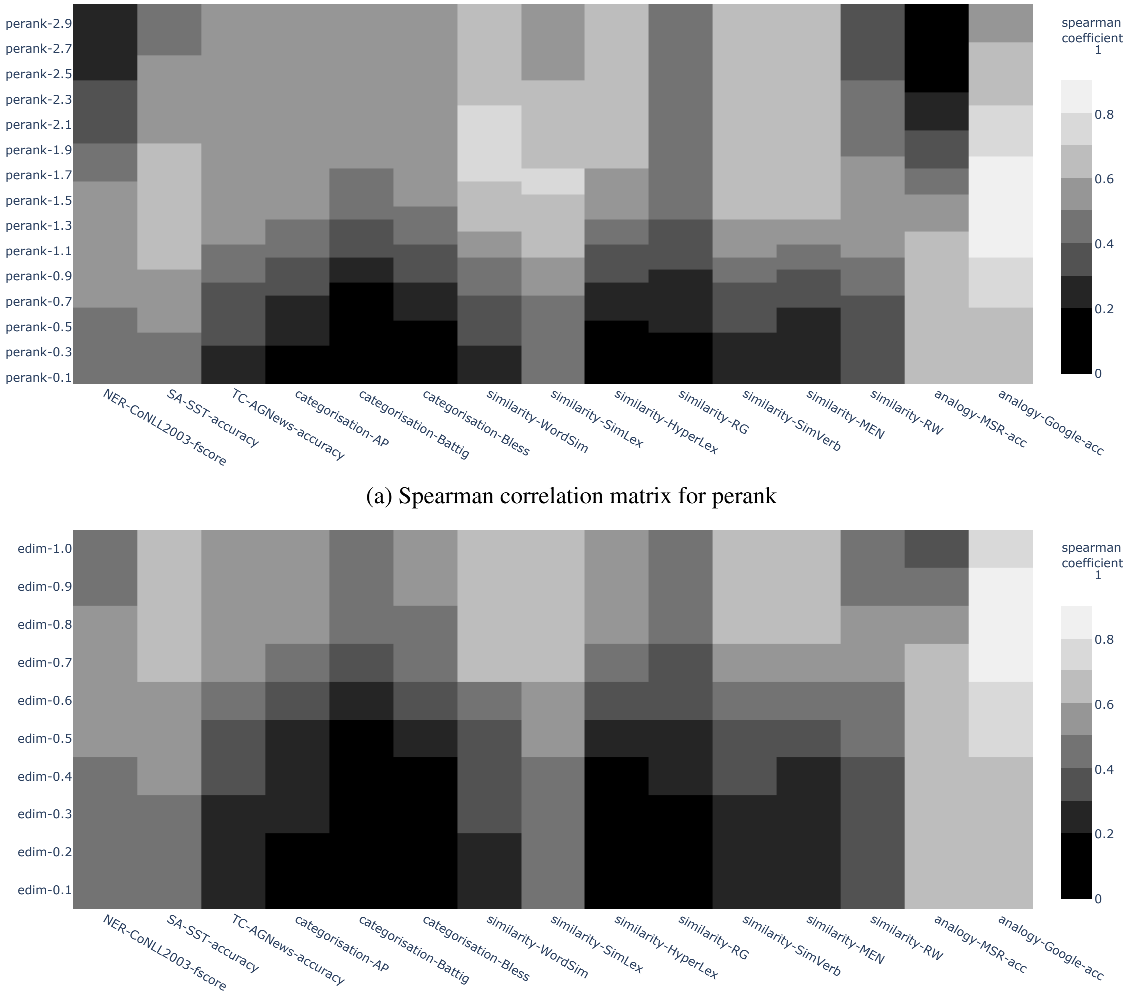


Figure 2: Pearson correlation matrix between extrinsic and intrinsic evaluations. The first three columns are extrinsic tasks, while the rest are intrinsic ones. For each evaluation, we indicate the category of evaluation followed by the dataset. For extrinsic task, we eventually add the aggregation metric (Fscore or Accuracy).



(a) Spearman correlation matrix for perank

(b) Spearman correlation matrix for edim

Figure 3: perank (3a) and edim (3b) correlation matrices with other evaluations. Each row corresponds to a different value of p and is labelled in the following format $\mathrm{perank}-\{p\}$ and $\mathrm{edim}-\{p\}$.

The objective of this part is to investigate correlations between those scores.

For clarity, we divided the results into four different figures:

- Figure 2 summarises the Pearson correlation coefficient between each pair of extrinsic or intrinsic evaluations. Pearson coefficient is chosen here because we assume the dependence between scores to be linear: the improvement of a specific task must be proportional to other task improvements.
- Figure 3a gives Spearman correlation coefficient between powered effective rank and extrinsic/intrinsic evaluations. Spearman is preferred instead of Pearson since the correlation between global metrics and evaluations is potentially not proportional. We do not report correlations using GIDM since those metrics do not correlate well with other evaluations.
- Figure 3b gives Spearman correlation coefficient between powered effective rank and extrinsic/intrinsic evaluations.
- Figure 4 reports the scores for each intrinsic and extrinsic task explained by global metrics (edim and perank, for values of p leading to the best correlation scores, and low correlation with the embedding dimension).

4.4. Synthesis

Based on our results, we derived three main remarks.

Task / Dataset independence. Figure 2 shows linear correlations between pair of tasks. As visible on this figure, a large number of intrinsic tasks are strongly correlated (coefficient > 0.9). 7 tasks seem remarkably independent from others (from left to right): NER, SA, TC, categorisation-Battig, similarity-RW, analogy-MSR and analogy-Google. An explanation for this is that those tasks catch language aspects not handled by other evaluations. For instance similarity-RG and similarity-Wordsim are particularly linearly dependent since they assess the same notion: similarity of common words. At the opposite, similarity-RW and similarity-Wordsim are not as dependent since RW essentially contains infrequent words. With such figures, we can constitute a set of tasks assessing independent aspects of language, avoiding redundancy. In practice, we should avoid measuring redundant information and focus on evaluations catching distinct language aspects. In doing this, we would obtain a more accurate picture of embedding qualities.

Fast selection. A common problem in downstream tasks is hyper-parameters optimisation. This step is time-consuming and often ignores the optimisation of word embedding parameters. Indeed, it focuses only on the hyper-parameters of the downstream model. Figure 2, 3a and 3b expose moderate correlations between intrisic/global evaluations and extrinsic tasks. This result is important since intrinsic and global evaluations are faster to carry out than extrinsic ones. Therefore, considering a set of word embeddings trained with different hyper-parameters, they can help choosing the word embedding likely to yield the best results. This seems confirmed by Figure 4: best performances are obtained for the highest values of global metrics (perank and edim). However, we must admit that intrinsic / global evaluations are only indicators pointing toward the best word embedding. If possible, one must still prefer optimising word embeddings with regard to the final downstream objective, as shown by Claveau and Kijak (2016) and Schnabel et al. (2015).

Optimisation criterion. For certain values of p, powered effective rank and empirical dimension are positively correlated with most of evaluations, as shown in Figure 3a and 3b. This implies that maximising perank / edim would simultaneously increases scores of other evaluations. This point is also suggested on Figure 4. However, we saw in Section 2 that maximising perank / edim is equivalent to equalising singular values. This means that the word embedding matrix should be orthogonal or close to an orthogonal matrix. Consequently, it would be beneficial to regularise this matrix such that it cannot be far from being orthogonal. This could be achieved using the SRIP regulariser proposed by Bansal et al. (2018), or SVD parameterisation as in the work of Zhang et al. (2018) and Arora et al. (2019). The major problem is the size of the word embedding matrix which makes the optimisation process time-consuming.

Actually, the points on optimisation criterion and fast selection are closely related. They both stand in favour of the maximisation of edim or perank. As shown in Figure 3 and 4, it seems crucial to have the highest edim (or perank) possible in order to perform well on intrinsic or extrinsic tasks.

5. Conclusion

In this work, we created and evaluated a large variety of FastText embeddings. From these experiments we outlined significant correlations between all kinds of evaluations. Empirical dimension is a global metric taken from computer vision. This one helped us to discover the necessity of orthogonal regularisation. Indeed, a high empirical dimension seem to positively influence the performance on various intrinsic and extrinsic evaluations. Therefore, maximising the empirical dimension while learning word embeddings should improve its downstream effectiveness.

In addition to edim, we defined the *powered effective rank* (perank), an extension of the effective rank introducing a parameter controlling the orthogonal sensitivity. The empirical dimension was already proposing to control that aspect. However, the perank is defined on a larger domain and, thus, we expect it to discover new regions hidden by the domain constraints of the empirical dimension. We observed that perank is less sensitive to the embedding dimension for high values of p than edim. Thus, a criterion based on perank may be more adapted to regularise intrinsic vector structures, than a criterion based on edim.

This study exposes the complexity of evaluation and its importance. Our experiments probed task independence. This is an important point since one prefers to assess a model with a set of independent tasks in order to obtain a big and complete picture of the quality of its model. Considering our

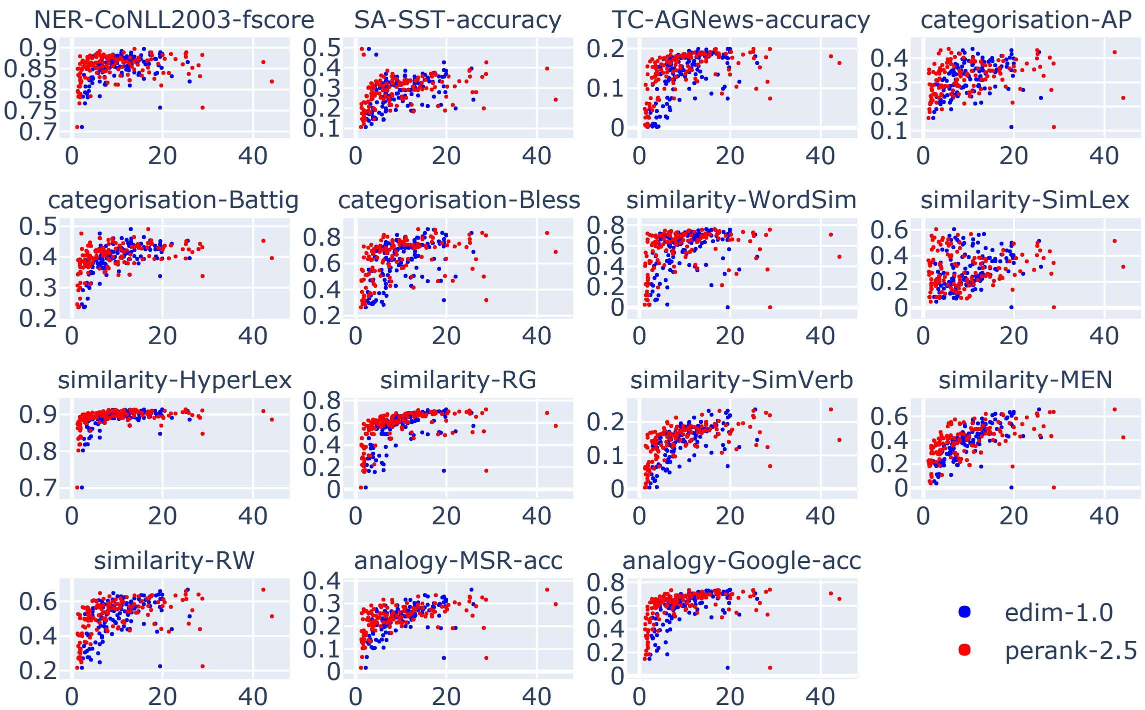


Figure 4: Scores (vertical axis) for all intrinsic and extrinsic tasks explained by edim or perank (horizontal axis), respectively with $p = 1$ or $p = 2.5$. Each point corresponds to a word embedding and represents its performance on extrinsic or intrinsic tasks (y-axis) and its edim or perank value (x-axis).

work and the state of the art, correlations seem to be significant if underlying embeddings are trained with an identical algorithm and different parameters. Future investigations may try to use global metrics as regularisation terms during the learning process and observe whether it improves correlated extrinsic evaluations. Another important future study is to apply this methodology to other categories of algorithm. As we only studied FastText here, it would be essential to see if our work generalises to other word embedding techniques.

6. References

Almuhareb, A. (2006). Attributes in lexical acquisition.

Amsaleg, L., Chelly, O., Furon, T., Girard, S., Houle, M., Kawarabayashi, K.-i., and Nett, M. (2015). Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 29–38, New York, NY, USA. ACM.

Arora, S., Cohen, N., Hu, W., and Luo, Y. (2019). Implicit regularization in deep matrix factorization. *CoRR*, abs/1905.13655.

Bakarov, A. (2018). A survey of word embeddings evaluation methods. *CoRR*, abs/1801.09536.

Bansal, N., Chen, X., and Wang, Z. (2018). Can we gain more from orthogonality regularizations in training deep cnns? *CoRR*, abs/1810.09102.

Baroni, M. and Lenci, A. (2011). How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, July. Association for Computational Linguistics.

Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive science*, 34:222–54, 03.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, January.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Claveau, V. and Kijak, E. (2016). Direct vs. indirect evaluation of distributional thesauri. In *International Conference on Computational Linguistics, COLING*, Osaka, Japan, December.

Claveau, V. (2018). Indiscriminateness in representation spaces of terms and documents. In *ECIR 2018 - 40th European Conference in Information Retrieval*, volume 10772 of *LNCS*, pages 251–262, Grenoble, France, March. Springer.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, January.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. S. (2017). Allennlp: A deep semantic natural language processing platform.

Gerz, D., Vulic, I., Hill, F., Reichart, R., and Korhonen, A.

(2016). Simverb-3500: A large-scale evaluation set of verb similarity. *CoRR*, abs/1608.00869.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *American Journal of Computational Linguistics*, 41(4):665–695, December.

Houle, M., Kashima, H., and Nett, M. (2012). Generalized expansion dimension. pages 587–594, 12.

Joshi, M., Prajapati, P., Shaikh, A., and Vala, V. (2017). A survey on sentiment analysis. *International Journal of Computer Applications*, 163:34–38, 04.

Karypis, G. (2003). Cluto: A clustering toolkit. *Technical Report 02-017, University of Minnesota (Department of Computer Science).*

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., and Brown, D. E. (2019). Text classification algorithms: A survey. *CoRR*, abs/1904.08067.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Li, J., Sun, A., Han, J., and Li, C. (2018). A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria, August. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mikolov, T., Yih, W.-t., and Zweig, G. (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, Atlanta, Georgia, June. Association for Computational Linguistics.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Poling, B. and Lerman, G. (2013). A new approach to two-view motion segmentation using global dimension minimization. *CoRR*, abs/1304.2999.

Qiu, Y., Li, H., Li, S., Jiang, Y., Hu, R., and Yang, L. (2018). Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In Maosong Sun, et al., editors, *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 209–221, Cham. Springer International Publishing.

Roy, O. and Vetterli, M. (2007). The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610, Sep.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October.

Sang, E. F. T. K. and Meulder, F. D. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, September. Association for Computational Linguistics.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Tifrea, A., Becigneul, G., and Ganea, O.-E. (2018). PoincarÃ© glove: Hyperbolic word embeddings.

Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., and Dyer, C. (2015). Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal, September. Association for Computational Linguistics.

Vulić, I., Gerz, D., Kiela, D., Hill, F., and Korhonen, A. (2017). HyperLex: A large-scale evaluation of graded lexical entailment. *American Journal of Computational Linguistics*, 43(4):781–835, December.

Zhang, J., Lei, Q., and Dhillon, I. S. (2018). Stabilizing gradients for deep neural networks via efficient SVD parameterization. *CoRR*, abs/1803.09327.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *CoRR*, abs/1611.06639.