Low-rank approximations of second-order document representations

Jarkko Lagus
Reaktor Innovations Oy,
University of Helsinki
jalagus@cs.helsinki.fi

Janne Sinkkonen
Reaktor Innovations Oy
janne.sinkkonen@reaktor.fi

Arto Klami
University of Helsinki
Department of Computer Science
arto.klami@cs.helsinki.fi

Abstract

Document embeddings, created with methods ranging from simple heuristics to statistical and deep models, are widely applicable. Bag-of-vectors models for documents include the mean and quadratic approaches (Torki, 2018). We present evidence that quadratic statistics alone, without the mean information, can offer superior accuracy, fast document comparison, and compact document representations. In matching news articles to their comment threads, low-rank representations of only 3–4 times the size of the mean vector give most accurate matching, and in standard sentence comparison tasks, results are state of the art despite faster computation. Similarity measures are discussed, and the Frobenius product implicit in the proposed method is contrasted to Wasserstein or Bures metric from the transportation theory. We also shortly demonstrate matching of unordered word lists to documents, to measure topicality or sentiment of documents.

1 Introduction

Today, most computational models for natural language are based on distributional representations. Words are routinely represented by *word embeddings* (Mikolov et al., 2013), most commonly as fixed-dimensional real-valued vectors, such as GloVe (Pennington et al., 2014) and fastText (Mikolov et al., 2018). Even though there is extensive literature on using, e.g., character-level and other sub-word information (Lee et al., 2017; Radford et al., 2017; Mikolov et al., 2018) or non-Euclidean embedding spaces (Nickel and Kiela, 2017; Muzellec and Cuturi, 2018), the standard embeddings remain currently as the default building block for practical tools.

Most applications do not, however, care about individual words. Instead, we may be concerned about the meaning of sentences or retrieval of documents, or in general, units larger than words. Distributed representations can be built for these larger units as well. Although sentences and documents differ as linguistic concepts, computational models for them can be similar when they are considered as sequences or even (unordered) bags of words.

Document representations often build on word embeddings. Already the mean of the word vectors turns out to be a surprisingly good representation (Wieting et al., 2015b), and accounting for the importance of words by a weighting scheme improves it further (Arora et al., 2016; Gupta et al., 2019). Even though the bare mean clearly ignores information, it is very efficient to compute. On the other end of the spectrum, document embeddings are built with computationally extremely heavy deep learning models such as ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), and BERT (Devlin et al., 2018). Deep models produce rich representations, but the amount of data and computation needed for training make them prohibitive for many applications.[1]

Our work falls between these two extremes. With the understanding that mean vectors may miss important aspects of documents, we want to develop fast and easy-to-use tools. This rules out complex deep networks. Instead, we focus on using second-order interactions between words, building on *covariance* of the embeddings of individual words, following the recent works of Torki (2018) and Nikolentzos et al. (2017).

The motivation of the paper is on finding fast and accurate ways to compare documents, or, alternatively, documents and semantics spanned by word lists. We start by evaluating document sim-

[1] For example BERT takes 0.5 secs to process a sentence on a CPU (Nvidia blog, our experiments), and getting good document representations may require fine-tuning.

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 634–644
Hong Kong, China, November 3-4, 2019.

ilarity as pairwise similarities between words and show that this induces a compact approximative representation for the documents themselves. We relate the pairwise similarity to Wasserstein or Bures metric, used recently in various machine learning tasks (Arjovsky et al., 2017; Muzellec and Cuturi, 2018) and in quantum information theory (Bhatia et al., 2018).

The main result of these derivations is a practical document embedding strategy that builds on pre-trained word embeddings. The document embeddings are of relatively low dimension, larger than the word embeddings only by a small factor. They allow for efficient comparisons and are easy to implement and use in downstream tasks of document retrieval, sentence classification, etc. We demonstrate competitive performance against state-of-the-art methods in standard sentence similarity tasks (Conneau and Kiela, 2018), with a lower computational cost. We further demonstrate the approach in matching articles to their comment chains, and briefly in scoring moral sentiment and topicality defined by word lists.

2 Related work

Work on document representations has a long history in information retrieval. Sentence embeddings (Arora et al., 2016; Perone et al., 2018) is a related topic that has lately become more prominent, maybe because of the fast growth of social media platforms where communication is mostly done via short messages. For this paper, we treat sentences as short documents.

The mainline of research deals with building document vectors from pre-trained word vectors. The straightforward way averages over the word vectors. Wieting et al. (2015b) show that complex computations are not necessary for good document vectors. Instead, smart weighting under the averaging model is usually sufficient. On top of that work, weighting schemes and other heuristics have been proposed. The latest include common component removal (Arora et al., 2016), and the weighting schemes SIF (Arora et al., 2016), and P-SIF (Gupta et al., 2019), similar in idea to TF-IDF weighting. As alternatives to usage of pre-trained word embeddings, one can train directly document embeddings like skip-thought (ST) vectors (Kiros et al., 2015) basically generalizing the *word2vec* training method to sentences, or train the word embeddings and document embeddings together, but still within the bag-of-the-words averaging framework as is done with Paragraph Vectors (Le and Mikolov, 2014) and Doc2VecC (Chen, 2017).

Our core contributions are in the use of second-order information—covariance of the word vectors—for improving the representations. To our best knowledge, there is quite limited previous work in this direction. Torki (2018) used covariance matrices as (quite high-dimensional) representation for documents and Nikolentzos et al. (2017) represented documents with Gaussian distributions and used divergence metrics to compare the imposed distributions. We provide technical and computational analysis of the covariance approach, discuss similarity measures for the representations, including Frobenius and Wasserstein inner products, and show how low-rank approximations can then speed up the comparisons and make the representations more compact.

A completely different approach is taken by the deep learning community, with the use of universal language and transformer models such as ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), and BERT (Devlin et al., 2018). The accuracy of these deep learning models is state of the art, but the computational cost and need for training data are high. At the time of writing, there are no pre-trained models available for the Finnish language used in our experiments, and training such a base model would be costly as shown by Strubell et al. (2019).

3 Representations and similarities

Most applications compare word embeddings by the cosine similarity, $\cos(w_1, w_2) = \frac{w_1^T w_2}{|w_1||w_2|}$, where $w_1, w_2 \in \mathcal{R}^d$ are the embeddings (vectors). Cosine similarity is invariant to lengths of the vectors. Lengths typically do not encode semantics but relate to aspects like frequency of the word or homogeneity of its context.

Documents or sentences (later simply documents) are, in the absence of a sequence model, treated as bags of words. That is, methods for comparing documents are invariant to word order. We define for later purposes a *document matrix*,

$$D = \begin{bmatrix} \cdots & w_1 & \cdots \\ \cdots & w_2 & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & w_n & \cdots \end{bmatrix} \in \mathcal{R}^{n \times d} \,, \qquad (1)$$

as a collection of word vectors. Although the representation as such is not order-invariant, order-invariant ways to compare documents can be derived for this representation. The simplest is the cosine similarity of means, $\cos(1^T D_1/n_1, 1^T D_2/n_2)$. Performance of document mean vectors in benchmarks is not quite a state of the art, but decent enough for many applications (Wieting et al., 2015b).

A refinement from the simple average is to reweigh the word vectors before averaging, either in a general way or by specializing into the current corpus. Term-frequency inverse-document-frequency (TF-IDF) weighting is the classic way. Later variations of weighting schemes are smooth inverse frequency (SIF) (Arora et al., 2016) and its derivative partition SIF (P-SIF) (Gupta et al., 2019). They use weighted mean with weights computed from corpus; SIF uses $\alpha/(\alpha + p(w))$, where $\alpha > 0$ controls the smoothing based on empirical probability $p(w)$ of word w. The precomputation of weights for the whole corpus makes SIF unusable in some cases where the whole corpus is not available or is extremely large, and prohibits online processing. The issue is even more severe with P-SIF that requires more elaborate preprocessing.

For further improvement within the mean vector framework, it should be possible to improve performance either by (a) computing word embeddings in a way that optimizes the document embeddings (still computed as word vector averages), or by (b) transformations of the document averages in a way that takes the current document corpus into account. An example of the former is the Doc2VecC embedding (Chen, 2017), and SIF (Arora et al., 2016) demonstrates the latter by removing variation common to the corpus by projecting away the main variation over documents.

Still, a third way to improve performance would be to expand the representation from average while maintaining order invariance for model simplicity. This leads to second-order representations discussed next.

4 Second-order document representations

Our motivation arises from an empirical observation that mean vectors are not necessarily efficient summaries of documents (Fig. 1). At least with the *word2vec* embeddings, the distribution of word vectors (on the 2D-space of their principal variance) is strongly skewed, with a comet-like shape, often with more than one separate ”tails”. Mean is not an efficient statistic for such distribution, and it is an especially poor description of the tails that seem to carry the most descriptive words.

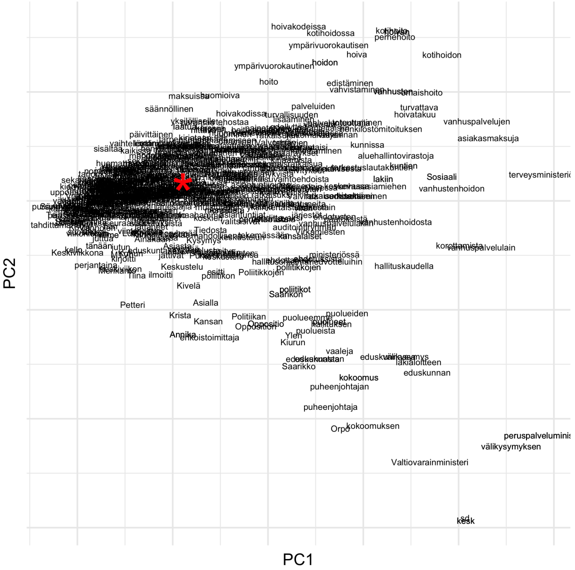

Figure 1: Words of a Finnish news article as *word2vec* vectors, projected to the 2D space of principal variation. The broad tail up right contains words related to health care and social services, especially those of seniors. The lower right tail is about politics. The mean vector (red star) makes a compromise between the tails, and is determined to a large degree by the numerous non-descriptive words projected closer to the origin. A representation that takes higher moments into account would catch the tails better. This structure seems to be typical to most of the documents in our corpora.

A representation based on higher-order moments would better catch the characteristics of the word vectors. Second moments, or covariance if centered to the means, would lose the asymmetry of the distributions, but the advantages compared to still higher-order moments are (relative) compactness and elegant algebra.

A question remains how second-order representations should be compared pairwise. A related worry is the computational efficiency, as the computational cost for naive second-order representations scales quadratically with the dimensionality of the embedding. Below, we first present an extension of cosine similarity to pairs of unordered collections of word vectors, and note how this induces a second-order representation for documents, and is interpretable as a Frobenius inner product. We note a connection of this approach to existing work, compare to Bures and Wasserstein metric (later referred only as Wasserstein) and optimal transport theory, and later in the paper present empirical evidence of good performance of low-rank approximations of these representations.

4.1 Extending cosine similarity to documents

Using the notation of (1), let $A \in \mathcal{R}^{n\times d}, B \in \mathcal{R}^{m\times d}$ be the word vectors of two documents, sentences, or other unordered collections of words, with word counts (n, m) and embedding of dimensionality d.

The core of cosine similarity of single words a, b is the ordinary inner product $\cos(a, b) \propto a^T b$, with a normalization such that $\cos(a, a) = 1$ for any $a \neq 0$. We generalize this to similarities of word collections A and B, as the normalized square sum of all pairwise dot products:

$$\begin{aligned} Z(A,B)\, h^2(A, B) \equiv& \\ &\sum_{a,b}(a^T b)^2 = \sum_{a,b} ab^T\, ba^T \\ =&\sum_a aB^T Ba^T = \sum_a \mathrm{Tr}(Ba^T aB^T) \\ =&\,\mathrm{Tr}(BA^T AB^T) = \mathrm{Tr}(A^T AB^T B) \end{aligned}$$

with $Z(A, B)$ such that $h^2(A, A) = 1$ for any $A \neq 0$. The unnormalized trace structure appears so often later in the paper, that we define a shorter notation for it:

$$A * B \equiv [\mathrm{Tr}(BA^T AB^T)]^{1/2}. \tag{2}$$

The trace is interpretable as the *Frobenius inner product* of covariance-like but unnormalized matrices $C_A \equiv A^T A$ and $C_B \equiv B^T B$:

$$(A * B)^2 = \mathrm{Tr}(C_A^T C_B)\,. \tag{3}$$

With normalization, the similarity then becomes

$$h^2(A, B) = \frac{(A * B)^2}{(A * A)\,(B * B)} \equiv A' * B'\ , \tag{4}$$

where the last expression prenormalizes the word list matrices so that

$$A' \equiv A/(A * A)^{1/2}\,. \tag{5}$$

Finally, one can centralize word matrices before computing the similarity, $A_0 = A - 1^T m_A$, etc., without affecting the formalism:

$$h(A_0, B_0) = A'_0 * B'_0\,. \tag{6}$$

Torki (2018) introduced a document representation named DoCoV and extended it in experiments to include mean vector gaining an increase in performance. This variant of the DoCoV representation is defined as $\mathrm{vec}(m_A, A_0^T A_0)$, where the vec operation concatenates the arguments and flattens the matrices row by row. This is similar to our cross product C_A but includes the mean. For computing similarities they used dot products

$$\begin{aligned} m_A^T m_B &+ vec(A_0^T A_0)^T\, vec(B_0^T B_0) \\ &= m_A^T m_B + (A_0 * B_0)^2\,. \end{aligned}$$

We note that this is the similarity h^2 introduced above, summed to the cosine similarity of the means, but the normalization of the dot product by Torki (2018) is for the entire concatenated vectors, which is reasonable as long as mean and covariance are treated as commensurable. There is a unit inconsistency in the concatenation, for the covariance term is quadratic while the mean is not. Our reinterpretation and slight reformulation of the similarity as h^2 offers a way of substantially shrinking the document representations (Section 5), and including the mean does not seem empirically important.

4.2 Connection to transport theory

Wasserstein metric compares two (elliptic or gaussian) distributions defined by their means and covariances. It emerges in the optimal transport theory as a measure of minimum-path transport of the probability mass of distribution (m_A, C_A) to distribution (m_B, C_B):

$$\begin{aligned} &\mathcal{W}^2((m_A, C_A), (m_B, C_B)) = \\ &\quad ||m_A - m_B||^2 + \\ &\quad \mathrm{Tr}\left(C_A + C_B - 2\left(C_A^{1/2}\, C_B\, C_A^{1/2}\right)^{1/2}\right)\,. \end{aligned}$$

For covariances computed from word vectors, $C_A = A^T A$, etc., we would like to have scale invariance similar to cosine or the Frobenius cosine above. But it is not clear how to obtain scale invariance in a principled way, for the scales of terms $\mathrm{Tr}\, C_A$ and $\mathrm{Tr}\, C_B$ vary differently from the scale of the cross term.

Within the context of their elliptical embeddings, Muzellec and Cuturi (2018) define a Bures or Wasserstein cosine by normalizing the sole cross term[2]:

$$h_{\mathcal{W}}(C_A, C_B) = \frac{\mathrm{Tr}\left(C_A^{1/2}\, C_B\, C_A^{1/2}\right)^{1/2}}{[\mathrm{Tr}\, C_A]^{1/2}\, [\mathrm{Tr}\, C_B]^{1/2}} \, .$$

This form is surprisingly close to the "Frobenius cosine" of Eq. 4, and allows prenormalization of representations: First by applying the cyclic property of trace, and then moving normalizations to be computed first, we have

$$\begin{aligned} h_{\mathcal{W}}(C_A, C_B) &= \frac{\mathrm{Tr}\left(C_A^T\, C_B\right)^{1/2}}{[\mathrm{Tr}\, C_A]^{1/2}\, [\mathrm{Tr}\, C_B]^{1/2}} \\ &= \mathrm{Tr}\left(\left[\frac{C_A}{\mathrm{Tr}\, C_A}\right]^T \left[\frac{C_B}{\mathrm{Tr}\, C_B}\right]\right)^{1/2} \, . \end{aligned}$$

The difference to our similarity defined in Eq. 4 is only in the order of the outer square root and trace operators. If one denotes the eigenvalues of the suitably normalized matrix $C_A C_B$ by η_i^2, the Wasserstein cosine is equal to $\sum_i \eta_i$, while the Frobenius cosine equals to $(\sum_i \eta_i^2)^{1/2}$. The latter obviously gives more weight to "high-variance co-variation" of C_A and C_B, that is, for larger eigenvalues. When only one eigenvalue is non-zero, the cosines are equal.

Matrix square root in the Wasserstein cosine, however, seems to require matrix diagonalization for every document comparison, which would be of computational complexity $O(d^3)$.

So although deriving anything equivalent to Frobenius cosine by starting from the Wasserstein metrics seems hard, similarities may be worth further investigation, either empirical or theoretical. An empirical comparison, presented in Figure 2, suggests that in practice the two cosines are quite closely related. Experiments later in the paper indicate better practical performance from the Frobenius cosine, in some applications. Frobenius cosine is notably faster to compute in practice, as shown in the next section.

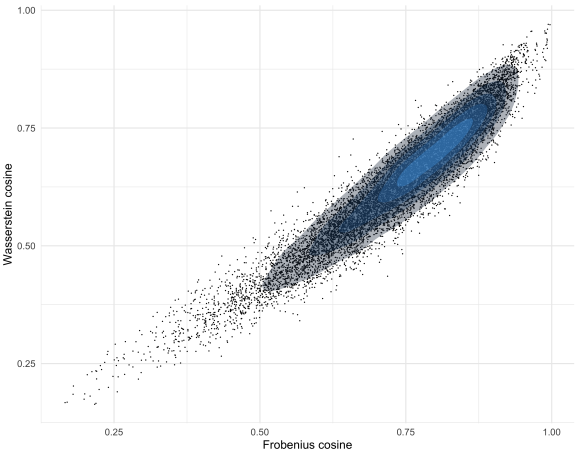

Figure 2: For our document collection of Finnish news articles and their comments, the Frobenius and Wasserstein cosines are closely related with correlation coefficient $r = 0.95$.

5 Computation and low-rank approximation

The Frobenius cosine of Eq. 4 can be computed by using word matrices A (of size $n \times d$) as document representations. Comparing these by pairwise vector inner products (Eq. 2) would have computationally cost of $O(n_A n_B d)$ — efficient for short word lists but not when lengths n approach or exceed d.

On the other hand, one can follow in the footsteps of Torki (2018) and represent documents as covariance-like inner products $A^T A$, which are of dimensionality $d(d+1)/2$. Torki (2018) named this method as DoCoV descriptor. In this case, the Frobenius complexity would then be $O(d^2)$, which for long documents is cheaper than a pairwise comparison of word lists, but still expensive compared to inner products of mean vectors, $O(d)$.

Key to low-order approximations is to note that the rows of the word lists A etc. do not need to represent words: Instead, any vectors $\hat{A}$ would give the same Frobenius product (and cosine) as long as the covariances are preserved, that is, $\hat{A}^T \hat{A} = A^T A$. If $\hat{A}$ is just an approximation of A, of, say, dimensionality $k \times d$ and of similar size for all documents, the pairwise Frobenius computation (Eq. 2) would be of complexity $O(k^2 d)$.

Our approximation and computation strategy is therefore to replace A with a suitable approximation $\hat{A}$, and compute the Frobenius inner product

[2] Or actually they have two separately normalized terms, one for means as ordinary cosine, and one for quadratics. We test the two-term version of our Frobenius product briefly in the experimental section and find that the mean term does not help performance there. The supplementary material of Muzellec and Cuturi (2018) easily leads to a similar conclusion with the Wasserstein cosine.

without ever realizing the covariance matrices, by

$$\hat{A} * \hat{B} = \left(\sum_{kl} (\hat{A})^T_{k\cdot} (\hat{B})_{l\cdot} \right)^{1/2} . \qquad (7)$$

For prenormalized approximations (Eq. 5), this is directly the desired similarity h^2 (as in Eq. 4).

We cannot optimize for approximation errors of pairwise Frobenius cosines, so we choose to optimize representations so that $A * A = \mathrm{Tr}(A^T A)$ is well preserved. Remembering that, for a symmetric matrix, trace is the sum of its eigenvalues, we may specifically choose a decomposition $H^T H$ of $A^T A$ such that dropping rows from H minimizes the approximation error in $\mathrm{Tr}(A^T A)$. The optimal H consists of the eigenvectors of $A^T A$ multiplied by square roots of their eigenvalues:

$$A^T A = U^T \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} U = (U\Lambda^{\frac{1}{2}})^T (U\Lambda^{\frac{1}{2}}) \equiv H^T H \, .$$

Now $\mathrm{Tr}(A^T A) = \sum_i \lambda_i$, where λ are the diagonal of Λ, the eigenvalues. All eigenvalues of a positive semidefinite matrix are non-negative, so trace is best approximated with a $\hat{H}$ that contains the eigenvectors associated to largest eigenvalues (the square roots of the eigenvalues multiplied in):

$$\hat{A} = \hat{H} = \begin{bmatrix} \sqrt{\lambda_1} u_1 \\ \sqrt{\lambda_2} u_2 \\ \cdots \\ \sqrt{\lambda_3} u_k \end{bmatrix} \in \mathcal{R}^{k \times d} \, . \qquad (8)$$

Let the SVD of the word list matrix be $A = U\Lambda' V^T$. Then $A^T A = V\Lambda' U^T U \Lambda' V^T = V\Lambda'^2 V^T$, the last form being the eigenvalue decomposition of $A^T A$. So the approximation $\hat{H}$ can be obtained directly from the SVD of the word list A, without computing the covariance matrix. This is relatively cheap for small ranks k, and can be scaled up for documents with very large n with stochastic methods (Halko et al., 2011). The complexity depends on the SVD algorithm chosen.

Note that analogously to the original word lists, the above approximation is applicable to centered word lists A_0, and the normalized counterparts A' and A'_0, as long as normalization of representations is done after approximation (to preserve $h^2(\hat{A}', \hat{A}') = 1$).

The Frobenius cosine can, therefore, be efficiently computed with Eq. 7 of complexity $O(k^2 d)$, if the documents are approximated by the first k principal vectors of the word list SVD. Savings, compared to full covariances, are of order k/d for space, and k^2/d for document comparison times.

6 Method summary

The pre-computation process for online comparison document comparison or search is as follows:

1. Choose suitable word embeddings for the corpus, defining factors being language and the domain.

2. Collect word vectors of each document into matrix A, as in Eq. 1.

3. Choose a rank k, typically 2–20. Compute the k-rank (SVD left side) approximation $\hat{A}_0$ of centered documents like in Eq. 8. Prenormalize (Eq. 5) and store $\hat{A}'_0$.

4. Compare documents by $\hat{A}'_0 * \hat{B}'_0$, using the pairwise dot products as in Eq. 7 for computation.[3]

Representations for new, upcoming documents repeat steps 2–3. (There is no global model to update or refer to.)

7 Experiments

To validate the proposed representations and the similarity h^2, we conducted two separate experiments and one demonstration. In the first one, proposed methods are compared to state-of-the-art sentence embedding models on the tasks of Facebook's SentEval library (Conneau and Kiela, 2018). The other experiment and the demonstration apply the techniques to news articles and their comments on three Finnish media sites. For the latter experiment, 88,986 articles with comments were gathered from the associated websites.

For English we use fastText (Mikolov et al., 2018) embeddings with $d = 300$, but note that similar results were achieved also with GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013). For Finnish, we use *word2vec* embeddings, provided by the Turku NLP group[4], since fastText does not provide adequate embeddings for Finnish.

[3] With rank $k = 1$, the similarity h^2 is equal to square of the cosine between directions of principal variation. The principal vectors have no well-defined polarity, so taking a square or absolute value of the cosine is important.

[4] Older version of the *fin-word2vec.bin* embeddings with dimension 300, linked from `http://bionlp.utu.fi/finnish-internet-parsebank.html`.

7.1 Textual similarity tasks

As an initial experiment on the quality of the second-order representations, we evaluated them with the standard unsupervised similarity tasks (STS12 - STS16) using SentEval library by Facebook (Conneau and Kiela, 2018).

We compare a few different low-rank approximations using our proposed method against the state-of-the-art unsupervised and semi-supervised methods in Table 1. We see that already at $k = 10$, the approximation reaches close to the accuracy of the full rank, within this dataset, while decreasing the size of representations significantly (here 300×10 vs. 300×300). It surpasses most of the other unsupervised methods and compares well with the semi-supervised methods. Interestingly, already the rank-1 approximation is better than the classical mean vector approach (Glove Avg in Table 1), while having the same computational complexity.

Our low-rank estimation is on par with the related unsupervised method, DoCoV, and the only model reaching clearly higher scores is the P-SIF + PSL by Gupta et al. (2019), which uses computationally relatively heavy reweighting.

Figure 3 demonstrates the validity of the low-rank approximation, by plotting the STS evaluation scores against the rank k of the approximation. The results are as expected: Using more components retains more information, and $k \approx 5 \ldots 10$ is roughly enough for all tasks. Contrary to the experiment of the next section, using too large k does not hurt performance (except nominally in some cases).[5]

7.2 Comments vs. article

Many news sites contain a comment section associated with articles. It can be useful to compare articles to the comments, for example for moderation: If the discussion drifts too far from the original topic, human attention may be needed. In this experiment, we tried to find real article–comment pairs from a set in which half of the pairs were fake, as a proxy for the moderation task. It tests the semantic resolution of the similarity measures and has the convenience of known ground truth.

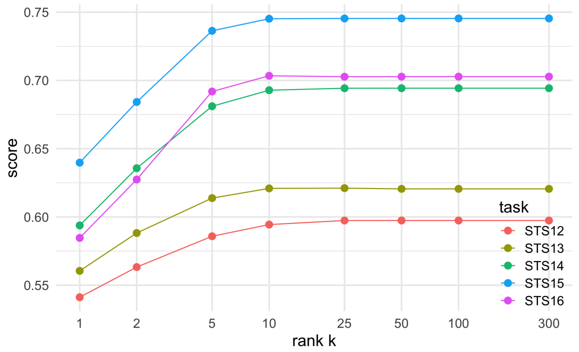

Figure 3: STS scores vs. rank k of the covariance approximation. Performance on these sentence tasks mostly saturates at $k = 5 \ldots 25$.

We crawled articles and their comment sections from three Finnish news sources: Yle (national broadcasting company), Uusi Suomi (news and blogging platform), and Iltalehti (a tabloid). Comments were concatenated to a single document (per article), retaining all article–comment pairs with at least 100 words on both sides. This gave 16,263, 18,548, and 9,682 pairs for Iltalehti, Uusi Suomi and Yle, respectively. The pairs were concatenated to sets of fake pairs of equal size, obtained by permuting the real pairs. We then rank the pairs according to decreasing similarity, and consider real pairs positive instances for retrieval measures. Figure 4 shows the ROC-like curves for our proposed h^2 of Eq. 7 (with rank $k = 4$), and two baselines based on document means, one computed directly from the word vectors and one after SIF weighting. The proposed method performs clearly better than average-based methods for all news sources, demonstrating the benefits of including second moments. It also outperforms the Wasserstein cosine $h_{\mathcal{W}}$ while having a clear advantage also in computation speed, and centering improves the results.

To see the effect of the rank, we ran the same test for various values of k and evaluated the recall at the selected value of precision (5% of fake pairs retrieved). All ranks $k > 0$ outperform the baseline of mean-vector cosine (horizontal line), except for Uusi Suomi at $k = 1$ (Figure 5). A bit unexpectedly, optimum is already at low values of k, which is nice from the computational perspective and suggests a regularization effect from SVD maybe worth of further study.

Finally, we wanted to see if adding a conventional mean term to the similarity computed with centralized second-order terms only helps with resolution, and ran the tests with various values of the weight α for the second-order term. One

[5] This may be because of the relative shortness of sentences here, vs. documents in the other experiment. The short sentences cannot even span a high-dimensional representation.

	Unsupervised					Semi-supervised			Ours		
		Glove	Glove		P-SIF	PSL	Glove	PSL	Frob.	Frob.	Frob.
Task	ST	Avg	tf-idf	DoCoV	PSL	Avg	WR	WR	$k = 1$	$k = 10$	$k = 300$
STS12	30.8	52.5	58.7	56.4	65.7	52.8	56.2	59.5	54.1	59.4	59.7
STS13	24.8	42.3	52.1	62.1	64.0	46.4	56.6	61.8	56.0	62.1	62.1
STS14	31.4	54.2	63.8	70.3	74.8	59.5	68.5	73.5	59.4	69.3	69.4
STS15	31.0	52.7	60.6	76.2	77.3	60.0	71.7	76.3	64.0	74.5	74.6
STS16	51.4	47.2	51.1	73.0	73.7	63.3	72.4	72.5	58.5	70.3	70.3

Table 1: Our method vs. other unsupervised and semi-supervised methods (from Gupta et al., 2019) on semantic textual similarity (STS) tasks, evaluated as Pearson correlations to human ground truth. ST stands for skip-thought vectors (Kiros et al., 2015), WR for SIF weighting combined with common component removal (Arora et al., 2016), and PSL for PARAGRAM-SL999 word vectors (Wieting et al., 2015a).

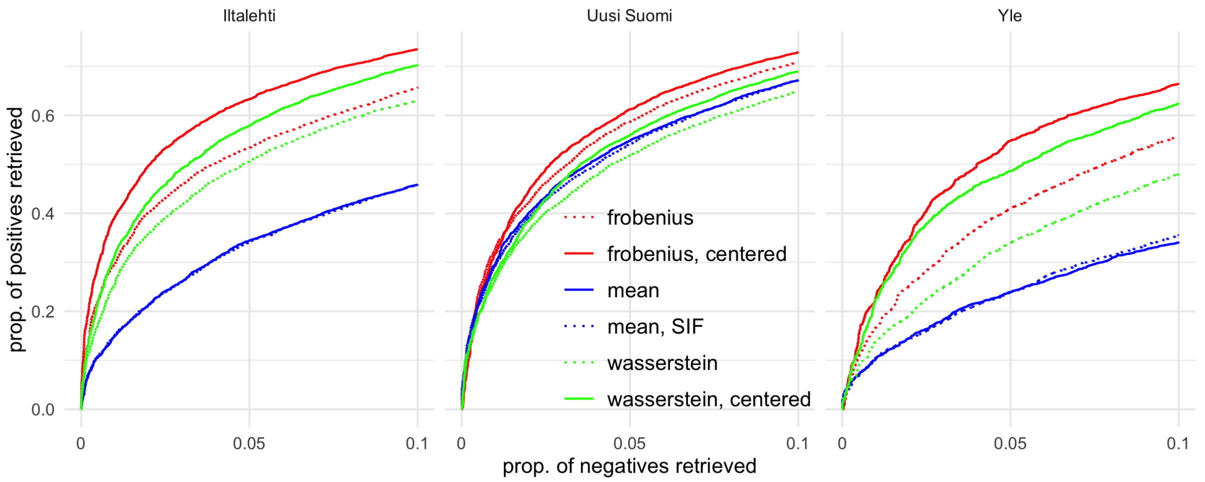

Figure 4: Fake vs. real pairs retrieved (ROC curves), in the article–comment matching tasks, for the three news sources, and for different matching methods. Cosine of mean vectors with original *word2vec* weights (blue; solid) and SIF weights (blue; dotted) perform practically identically. Quadratic approaches are all better, in general Frobenius (red) outperforming Wasserstein (green), and centered covariances (solid) outperforming non-centered ones (dotted). More fine-grained evaluations over k and mean-vector cosine mixing (Fig. 5 and 6) are run for 5% of the fake pairs retrieved (at the middle of x-axis).

source (Uusi Suomi) peaks around $\alpha = 0.5$, but for other sources, the performance increases monotonically with the weight of h^2. Apparently, mean vector is at least sometimes redundant if just low-rank second-order information is available.[6]

7.3 Media segmentation

Finally, the classic task of scoring documents for sentiment or topic by a word list is amenable to the application of the Frobenius similarity h^2. The target word list is usually just an unordered collection of words, although it may be weighted. Extensive sets of word lists are curated, for example, by LIWC. Likely, the semantics of such a list would sometimes be better operationalized by the suggested quadratic representation rather than by the list itself or its vectorized mean (with respect to an embedding).

As an example, we just present a single finding in Figure 7. The moral content of the comment chains of online news articles seems to vary by source, and climate change as a topic has differing effects, depending on the platform. The moral word lists were manually augmented and translated from the lists available from the developers of the Moral Foundations Theory (MFT)[7].

8 Conclusions

As already demonstrated by Torki (2018) and Nikolentzos et al. (2017), taking second moments of word vectors into document represen-

[6] The appendix of (Muzellec and Cuturi, 2018) gives similar impression for Wasserstein cosine: The performance is relatively flat over wide range of α and sometimes seems to increase monotonically.

[7] https://www.moralfoundations.org/othermaterials

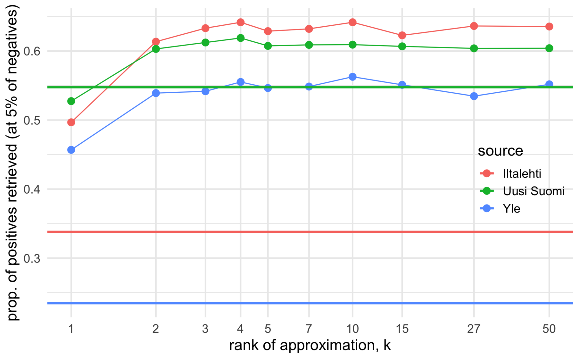

Figure 5: Frobenius recall of the article–comment matching task (proportion of real pairs retrieved when 5% of fake pairs are retrieved), as a function of rank k. For all news sources, performance saturates at rather low orders, $k = 2 \ldots 10$. Horizontal lines indicate recall with cosine of mean word vectors, which ignores second moments of the document.

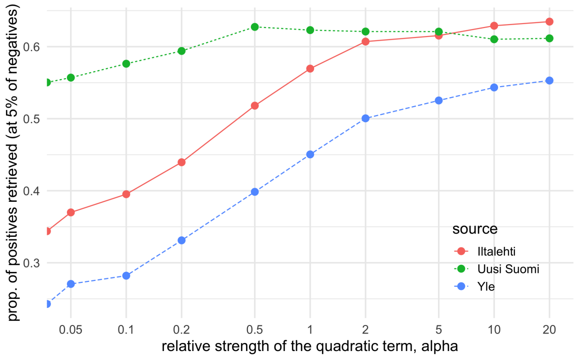

Figure 6: A recall measure on the article–comment matching task (proportion of real pairs retrieved when 5% of fake pairs are retrieved), when matching is with a mixture $\cos + \alpha h^2$ of ordinary cosine of means and centered Frobenius similarity (with rank $k = 3$). Adding mean information (smaller α) generally degrades performance except for one news source (*Uusi Suomi*) which peaks at $\alpha \approx 0.5$.

tations improves document matching. Surprisingly, the often-used mean vector seems to then become about irrelevant, a finding that needs to be replicated with other embeddings and larger experiments. There may exist efficient, ICA-like schemes relying on even higher moments to optimize the representations.

The second-order representations, and an associated similarity measure equivalent to the Frobenius inner product, can be derived by extending the Euclidean inner product into sets of words in a natural, pairwise manner. Empirically, the Frobenius similarity closely approximates Wasserstein similarity, familiar from transport theory, but al-

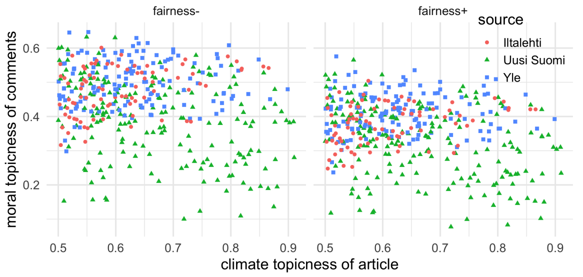

Figure 7: Moral sentiment of text, in the sense of the Moral Foundation Theory (Graham et al., 2013), has been measured by counting words appearing on a curated list (Garten et al., 2016). Moral word lists can also be related to documents with our h^2. On a blogging platform (*Uusi Suomi*), climate change as a prominent topic does not always rise moral comments at all (comments are probably technical). Points represent single article–comment pairs for articles of high topicality. Left: fairness, negative polarity; right: fairness, positive polarity.

lows efficient low-rank approximations of otherwise high-dimensional representations. The relationships of the two similarities may be worth further investigation, both theoretical and empirical.

Low-rank approximations are not only computationally useful, but also may sometimes have a regularizing effect that improves matching. Like mean vectors, second-order representations are useful as an alternative to traditional word-occurrence scoring, on quantifying sentiment and topicality of documents. Our experiments also show that rank-1 approximations are better representations of the documents than the mean vectors, while having the same representation size and computational complexity. There is an interesting contrast to the SIF preprocessing, where one step is to remove the corpus-wide largest component of variance to enhance the performance. While these two results are not contradictory, the combination is somewhat counter-intuitive.

Compared to other embedding methods, our approach requires only low precomputation effort and is local to document. The locality allows online processing that would be hard to implement with methods requiring preprocessing the corpus. If the online property is not needed, second-order representations are compatible with smart weighting schemes like SIF (Arora et al., 2016) or P-SIF (Gupta et al., 2019), and also with corpus-wide preprocessing schemes like projections and scalings.

References

Martin Arjovsky, Soumith Chintala, and Lon Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. 2018. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*.

Minmin Chen. 2017. Efficient vector representation for documents through corruption. *arXiv preprint arXiv:1707.02377*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Justin Garten, Reihane Boghrati, Joe Hoover, Kate M Johnson, and Morteza Dehghani. 2016. Morality between the lines: Detecting moral sentiment in text. In *Proceedings of IJCAI 2016 workshop on Computational Modeling of Attitudes*.

Jesse Graham, Jonathan Haidt, Sena Koleva, Matt Motyl, Ravi Iyer, Sean P Wojcik, and Peter H Ditto. 2013. Moral foundations theory: The pragmatic validity of moral pluralism. In *Advances in experimental social psychology*, volume 47, pages 55–130. Elsevier.

Vivek Gupta, Ankit Kumar Saw, Partha Pratim Talukdar, and Praneeth Netrapalli. 2019. Unsupervised Document Representation using Partition Word-Vectors Averaging.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Boris Muzellec and Marco Cuturi. 2018. Generalizing point embeddings using the wasserstein space of elliptical distributions. In *Advances in Neural Information Processing Systems*, pages 10258–10269.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347.

Giannis Nikolentzos, Polykarpos Meladianos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2017. Multivariate gaussian document representation from word embeddings for text categorization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 450–455.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Christian S. Perone, Roberto Silveira, and Thomas S. Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment. *CoRR*, abs/1704.01444.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. *arXiv preprint arXiv:1906.02243*.

Marwan Torki. 2018. A Document Descriptor using Covariance of Word Vectors. In *Proceedings of the*

56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 527–532.

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

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015b. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.