Word and Document Embedding with vMF-Mixture Priors on Context Word Vectors

Anonymous ACL submission

Abstract

Word embedding models typically learn two types of vectors: target word vectors and context word vectors. These vectors are normally learned such that they are predictive of some word co-occurrence statistic, but they are otherwise unconstrained. However, the words from a given language can be organized in various natural groupings, such as syntactic word classes (e.g. nouns, adjectives, verbs) and semantic themes (e.g. sports, politics, sentiment). Our hypothesis in this paper is that embedding models can be improved by explicitly imposing a cluster structure on the set of context word vectors. To this end, our model relies on the assumption that context word vectors are drawn from a mixture of von Mises-Fisher (vMF) distributions, where the parameters of this mixture distribution are jointly optimized with the word vectors. We show that this results in word vectors which are qualitatively different from those obtained with existing word embedding models. We furthermore show that our embedding model can also be used to learn high-quality document representations.

1 Introduction

Word embedding models are aimed at learning vector representations of word meaning (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017). These representations are primarily learned from co-occurrence statistics, where two words are represented by similar vectors if they tend to occur in similar linguistic contexts. Most models, such as Skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) learn two different vector representations $\mathbf{w}$ and $\tilde{\mathbf{w}}$ for each word w, which we will refer to as the target word vector and the context word vector respectively. Apart from the constraint that $\mathbf{w_i} \cdot \tilde{\mathbf{w}}_\mathbf{j}$ should reflect how often words w_i and w_j co-occur, these vectors are typically unconstrained.

As was shown in (Mu et al., 2018), after performing a particular linear transformation, the angular distribution of the word vectors that are obtained by standard models is essentially uniform. This isotropy property is convenient for studying word embeddings from a theoretical point of view (Arora et al., 2016), but it sits at odds with fact that words can be organised in various natural groupings. For instance, we might perhaps expect that words from the same part-of-speech class should be clustered together in the word embedding. Similarly, we might expect that organising word vectors in clusters that represent semantic themes would also be beneficial. In fact, a number of approaches have already been proposed that use external knowledge for imposing such a cluster structure, capturing the intuition that words which belong to the same category should be represented by similar vectors (Xu et al., 2014; Guo et al., 2015; Hu et al., 2015; Li et al., 2016c) or be located in a low-dimensional subspace (Jameel and Schockaert, 2016). Such models tend to outperform standard word embedding models, but it is unclear whether this is only because they can take advantage of external knowledge, or whether imposing a cluster structure on the word vectors is itself also inherently useful.

In this paper, we propose a word embedding model which explicitly aims to learn context vectors that are organised in clusters. Note that unlike the aforementioned works, our method does not rely on any external knowledge. We simply impose the requirement that context word vectors should be clustered, without prescribing how these clusters should be defined. To this end, we extend the GloVe model by imposing a prior on the context word vectors. This prior takes the form of a mixture of von Mises-Fisher (vMF) distributions, which is a natural choice for modelling clusters in directional data (Banerjee et al., 2005).

We show that this results in word vectors that are qualitatively different from those obtained using existing models, significantly outperforming them in syntax-oriented evaluations. Moreover, we show that the same model can be used for learning document embeddings, simply by viewing the words that appear in a given document as context words. We show that the vMF distributions in that case correspond to semantically coherent topics, and that the resulting document vectors outperform those obtained with existing topic modelling strategies.

2 Related work

A large number of works have proposed techniques for improving word embeddings based on external lexical knowledge. Many of these approaches are focused on external knowledge about word similarity (Yu and Dredze, 2014; Faruqui et al., 2015; Mrksic et al., 2016), although some approaches for incorporating categorical knowledge have been studied as well, as already mentioned in the introduction. What is different about our approach is that we do not rely on any external knowledge. We essentially impose the constraint that some category structure has to exist, without specifying what these categories look like.

The view that the words which occur in a given document collection have a natural cluster structure is central to topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and its non-parametric counterpart called Hierarchical Dirichlet Processes (HDP) (Teh et al., 2005) where the HDP model automatically discovers the number of latent topics based on the data characteristic. In recent years, several approaches that combine the intuitions underlying topic models with word embeddings have been proposed. For example, in (Das et al., 2015) it was proposed to replace the usual representation of topics as multinomial distributions over words by Gaussian distributions over a pre-trained word embedding, while (Batmanghelich et al., 2016) and (Li et al., 2016b) used von Mises-Fisher distributions for this purpose. Note that documents are still modelled as multinomial distributions of topics in these models. In (He et al., 2017) the opposite approach is taken: documents and topics are represented as vectors, with the aim of modelling topic correlations in an efficient way, while each topic is represented as a multinomial distribution over words. In this paper, we take a different approach for learning document vectors, by not considering any document-specific topic distribution. This allows us to represent document vectors and (context) word vectors in the same space and, as we will see, leads to improved empirical results.

Apart from using pre-trained word embeddings for improving topic representations, a number of approaches have also been proposed that use topic models for learning word vectors. For example, (Liu et al., 2015b) first uses the standard LDA model to learn a latent topic assignment for each word occurrence. These assignments are then used to learn vector representations of words and topics. Some extensions of this model have been proposed which jointly learn the topic-specific word vectors and the latent topic assignment (Li et al., 2016a; Shi et al., 2017). The main motivation for these works is to learn topic-specific word representations. They are thus similar in spirit to multi-prototype word embeddings, which aim to learn sense-specific word vectors (Neelakantan et al., 2014). Our method is clearly different from these works, as our focus is on learning standard word vectors (as well as document vectors).

Regarding word embeddings more generally, the attention has recently shifted towards contextualized word embeddings based on neural language models (Peters et al., 2018). Such contextualized word embeddings serve a broadly similar purpose as the aforementioned topic-specific word vectors, but with far better empirical performance. Despite their recent popularity, however, it is worth emphasizing that state-of-the-art methods such as ELMO (Peters et al., 2018) rely on a concatenation of the output vectors of a neural language model with standard word vectors. For this reason, the problem of learning standard word vectors remains an important research topic.

3 Model Description

The GloVe model (Pennington et al., 2014) learns for each word w a target word vector $\mathbf{w}$ and a context word vector $\tilde{\mathbf{w}}$ by minimizing the following objective:

$$\sum_{\substack{i,j \\ x_{ij}\neq 0}} f(x_{ij})(\mathbf{w_i} \cdot \tilde{\mathbf{w}}_\mathbf{j} + b_i + \tilde{b_j} - \log x_{ij})^2$$

where x_{ij} is the number of times w_i and w_j co-occur in the given corpus, b_i and $\tilde{b_j}$ are bias terms and $f(x_{ij})$ is a weighting function aimed at reducing the impact of sparse co-occurrence counts. It is easy to see that this objective is equivalent to maximizing the following likelihood function

$$P(D|\Omega) \propto \prod_{\substack{i,j \\ x_{ij} \neq 0}} \mathcal{N}(\log x_{ij}; \mu_{ij}, \sigma^2)^{f(x_{ij})}$$

where $\sigma^2 > 0$ can be chosen arbitrarily, and

$$\mu_{ij} = \mathbf{w_i} \cdot \tilde{\mathbf{w}}_\mathbf{j} + b_i + \tilde{b_j}$$

Here, D denotes the given corpus and Ω refers to the set of parameters learned by the word embedding model, i.e. the word vectors $\mathbf{w_i}$ and $\tilde{\mathbf{w}}_\mathbf{j}$ and the bias terms.

The advantage of this probabilistic formulation is that it allows us to introduce priors on the parameters of the model. This strategy was recently used in the WeMAP model (Jameel et al., 2019) to replace the constant variance σ^2 by a variance σ_j^2 that depends on the context word. In this paper, however, we will use priors on the parameters of the word embedding model itself. Specifically, we will impose a prior on the context word vectors $\tilde{\mathbf{w}}$, i.e. we will maximize:

$$\prod_{\substack{i,j \\ x_{ij} \neq 0}} \mathcal{N}(\log x_{ij}; \mu_{ij}, \sigma^2)^{f(x_{ij})} \cdot \prod_i P(\tilde{\mathbf{w}}_\mathbf{i})$$

Essentially, we want the prior $P(\tilde{\mathbf{w}}_\mathbf{i})$ to model the assumption that context word vectors are clustered. To this end, we use a mixture of von-Mises Fisher distributions. To describe this distribution, we begin with a von Mises-Fisher (vMF) distribution (Mardia and Jupp, 2009; Hornik and Grün, 2014), which is a distribution over unit vectors in $\mathbb{R}^d$ that depends on a parameter $\theta \in \mathbb{R}^d$, where d will denote the dimensionality of the word vectors. The vMF density for $\mathbf{x} \in \mathcal{S}_d$ (with $\mathcal{S}_d$ the d-dimensional unit hypersphere) is given by:

$$\mathit{vmf}(\mathbf{x}|\theta) = \frac{e^{\theta^\intercal \mathbf{x}}}{{}_0F_1(; d/2; \frac{\|\theta\|^2}{4})}$$

where the denominator is given by

$${}_0F_1(; p; q) = \sum_{n=0}^{\infty} \frac{\Gamma(p)}{\Gamma(p+n)} \frac{q^n}{n!}$$

which is commonly known as the confluent hypergeometric function. Note, however, that we will not need to evaluate this denominator, as it simply acts as a scaling factor. The normalized vector $\frac{\theta}{\|\theta\|}$, for $\theta \neq \mathbf{0}$, is the mean direction of the distribution, while $\|\theta\|$ is known as the concentration parameter. To estimate the parameter θ from a given set of samples, we can use maximum likelihood (Hornik and Grün, 2014).

A finite mixture of vMFs, which we denote as movMF, is a distribution on the unit hypersphere of the following form ($\mathbf{x} \in \mathcal{S}^d$):

$$h(\mathbf{x}|\Theta) = \sum_{k=1}^{K} \psi_k \, \mathit{vmf}(\mathbf{x}|\theta_k)$$

where K is the number of mixture components, $\psi_k \geq 0$ for each k, $\sum_k \psi_k = 1$, and $\Theta = (\theta_1, ..., \theta_K)$. The parameters of this movMF distribution can be computed using the Expectation-Maximization (EM) algorithm (Banerjee et al., 2005; Hornik and Grün, 2014).

Note that movMF is a distribution on unit vectors, whereas context word vectors should not be normalized. We therefore define the prior on context word vectors as follows:

$$P(\tilde{\mathbf{w}}) \propto h\Big(\frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|} \,|\, \Theta\Big)$$

Furthermore, we use L2 regularization to constrain the norm $\|\tilde{\mathbf{w}}\|$. We will refer our model as CvMF.

In the experiments, following (Jameel et al., 2019), we will also consider a variant of our model in which we use a context-word specific variance σ_j^2. In that case, we maximize the following:

$$\prod_{\substack{i,j \\ x_{ij} \neq 0}} \mathcal{N}(\log x_{ij}; \mu_{ij}, \sigma_j^2) \cdot \prod_i P(\tilde{\mathbf{w}}_\mathbf{i}) \cdot \prod_i P(\sigma_j^2)$$

where $P(\sigma_j^2)$ is modelled as an inverse-gamma distribution (NIG). Note that in this variant we do not use the weighting function $f(x_{ij})$, as this was found to be unnecessary when using a context-word specific variance σ_j^2 in (Jameel et al., 2019). We will refer this variant as CvMF(NIG).

Document embedding. The model described above can also be used to learn document embeddings. To this end, the target word vectors are simply replaced by document vectors and the counts x_{ij} then reflect how often word j occurs in document i. Below we will experimentally compare this strategy with existing methods for learning document representations, focusing especially on

Models	Gsem	GSyn	MSR	IM	DM	ES	LS
GloVe	78.85	62.81	53.04	55.21	14.82	10.56	0.881
SG	71.58	60.50	51.71	55.45	13.48	08.78	0.671
CBOW	64.81	47.39	45.33	50.58	10.11	07.02	0.764
WeMAP	**83.52**	63.08	55.08	56.03	14.95	**10.62**	0.903
CvMF	63.22	67.41	63.21	65.94	17.46	9.380	1.100
CvMF(NIG)	64.14	**67.55**	**63.55**	**65.95**	**17.49**	9.410	**1.210**

Table 1: Word analogy results on different datasets.

approaches that are inspired by probabilistic topic models. Indeed, we can intuitively think of the vMF mixture components in our model as representing topics. While there have already been topic models that use vMF distributions in this way (Batmanghelich et al., 2016; Li et al., 2016b), our approach is different because we do not consider a document-level topic distribution, and because we do not rely on pre-trained word embeddings.

4 Experiments

In this section we assess the potential of our model both for learning word embeddings (Section 4.1) and for learning document embeddings (Section 4.2). Our implementation will be made available upon acceptance.

4.1 Word Embedding Results

In this section, we describe the word embedding results, where we directly compare our model with the following baselines: GloVe (Pennington et al., 2014), Skipgram (Mikolov et al., 2013) (denoted as SG), Continuous Bag of Words (Mikolov et al., 2013) (denoted as CBOW), and the recently proposed WeMAP model (Jameel et al., 2019). We have used the Wikipedia dataset which was shared by Jameel et al. (2019), using the same vocabulary and preprocessing strategy. We report results for 300-dimensional word vectors and we use $K = 3000$ mixture components for our model. As evaluation tasks, we use standard word analogy and similarity benchmarks.

Analogy. Table 1 shows word analogy results for three datasets. First, we show results for the Google analogy dataset which is available from the GloVe project[1] and covers a mix of semantic and syntactic relations. These results are shown separately in Table 1 as *Gsem* and *Gsyn* respectively. Second, we consider the Microsoft syntactic word analogy dataset[2], which only covers syntactic relations and is referred to as *MSR*. Finally, we show results for the BATS analogy dataset[3], which covers four categories of relations: inflectional morphology (*IM*), derivational morphology (*DM*), encyclopedic semantics (*ES*) and lexicographic semantics (*LS*). The results in Table 1 clearly show that our model behaves substantially differently from the baselines: for the syntactic/morphological relationships (*Gsyn*, *MSR*, *IM*, *DM*), our model outperforms the baselines in a very substantial way. On the other hand, for the remaining, semantically-oriented categories, the performance is less strong, with particularly weak results for *Gsem*. For *ES* and *IS*, it needs to be emphasized that the results are weak for all models, which is partially due to a relatively high number of out-of-vocabulary words. In Figure 1 we show the impact of the number of mixture components K on the performance for *Gsem* and *Gsyn* (for the NIG variant). This shows that the underperformance on *Gsem* is not due to the choice of K. Among others, we can also see that a relatively high number of mixture components is needed to achieve the best results.

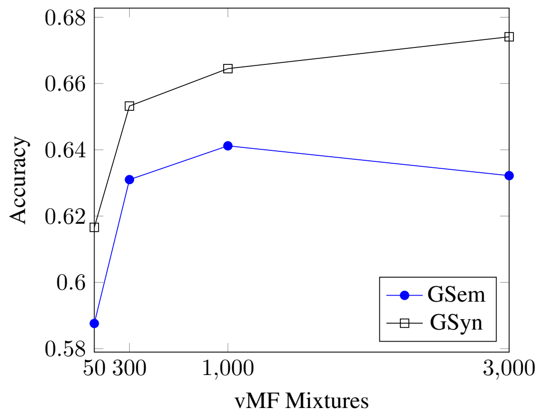

Figure 1: Accuracy vs number of vMF mixtures on the Google word analogy dataset for our model.

Word similarity. The word similarity results are shown in Table 2, where we have considered the same datasets as Jameel et al. (2019). In the table, we refer to EN-RW-Stanford as Stanf, EN-SIMLEX-999 as LEX, SimVerb3500 as Verb, EN-MTurk771 as Tr771, EN-MTurk287 as Tr287, EN-MENTR3K as TR3k, the RareWords dataset as RW, and the recently introduced Card-660 rare words dataset (Pilehvar et al., 2018) denoted as

[1]https://github.com/stanfordnlp/GloVe

[2]https://aclweb.org/aclwiki/Analogy_(State_of_the_art)

[3]http://vecto.space/projects/BATS/

Models	MC30	TR3k	Tr287	Tr771	RG65	Stanf	LEX	Verb143	WS353	YP130	Verb	RW	RW-660
GloVe	0.739	0.746	0.648	0.651	0.752	**0.473**	0.347	**0.308**	0.675	0.582	0.184	0.422	0.000
SG	0.741	0.742	0.651	0.653	0.757	0.470	0.356	0.289	0.662	0.565	0.195	0.470	0.000
CBOW	0.727	0.615	0.637	0.555	0.639	0.419	0.279	0.307	0.618	0.227	0.168	0.419	0.000
WeMAP	**0.769**	**0.752**	**0.657**	**0.659**	**0.779**	0.472	**0.361**	0.303	**0.684**	**0.593**	0.196	0.480	0.000
CvMF	0.707	0.703	0.642	0.652	0.746	0.419	0.353	0.250	0.601	0.465	**0.226**	**0.519**	0.000
CvMF(NIG)	0.708	0.703	0.642	0.652	0.747	0.419	0.354	0.250	0.604	0.467	**0.226**	**0.519**	0.000

Table 2: Word similarity results on some benchmark datasets.

RW-660. In most of these datasets, our model does not outperform the baselines, which is to be expected given the conclusion from the analogy task that our model seems specialized towards capturing morphological and syntactic features. Interestingly, however, in the *RW* dataset, which focuses on rare words, our model performs clearly better than the baselines. Intuitively, we may indeed expect that the use of a prior on the context words acts as a form of smoothing, which can improve the representation of rare words.

Qualitative analysis. To better understand how our model differs from standard word embeddings, Table 3 shows the ten nearest neighbors (Al-Rfou et al., 2013) for a number of words according to our CvMF(NIG) model and according to the GloVe model. What can clearly be seen is that our model favors words that are of the same kind. For instance, the top 5 neighbours of *fastest* are all speed-related adjectives. As another example, the top 7 neighbors of *red* are colors. To further explore the impact of our model on rare words, Table 4 shows the nearest neighbors for some low-frequency terms. These examples clearly suggest that our model captures the meaning of these words in a better way than the GloVe model. For example, the top neighbors of *casio* are highly relevant terms such as *notebook* and *compute*, whereas the neighbors obtained with the GloVe model seem largely unrelated. For comparison, Table 5 shows the nearest neighbors of some high-frequency terms. In these case we can see that the GloVe model obtains the best results, as e.g. *moreover* is found as a neighbor of *neural* for our model, and *indeed* is found as a neighbor of *clouds*. This supports the results from the similarity benchmarks that our model performs better than standard methods at modelling rare words but worse at modelling frequent words. Finally, Table 6 shows the effect that our model can have on ambiguous words, where due to the use of the prior, a different dominant sense is found.

4.2 Document Embedding Results

To evaluate the document embeddings, we focus on two downstream applications: categorization and document retrieval. As an intrinsic evaluation, we also evaluate the semantic coherence of the topics identified by our model.

Document Categorization. We have evaluated our document embeddings on four standard document classification benchmarks: 1) 20 Newsgroups (20NG)[4], 2) OHSUMED-23 (OHS)[5], 3) TechTC-300 (TechTC)[6], and 4) Reuters-21578 (Reu)[7]. As baselines, we consider the following approaches: 1) TF-IDF weighted bag-of-words representation, 2) LDA[8], 3) HDP[9], 4) the von Mises-Fisher clustering model (movMF)[10], 5) Gaussian LDA (GLDA)[11] and 6) Spherical HDP (sHDP)[12][13], 7) GloVe[14] (Pennington et al., 2014), 8) WeMAP (Jameel et al., 2019), 9) Skipgram (SG) and Continuous Bag-of-Words[15] (Mikolov et al., 2013) models. In the case of the word embedding models, we create document vectors in the same way as we do for our model, by simply replacing the role of target word vectors with document word vectors.

In all the datasets, we removed punctuation and non-ASCII characters. We then segmented the sentences using Perl. In all models, parameters

[4] http://qwone.com/ jason/20Newsgroups/

[5] https://www.mat.unical.it/OlexSuite/Datasets/SampleDataSets-download.htm

[6] http://techtc.cs.technion.ac.il/techtc300/techtc300.html

[7] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[8] https://radimrehurek.com/gensim/models/ldamodel.html

[9] https://github.com/blei-lab/hdp

[10] https://cran.r-project.org/web/packages/movMF/index.html

[11] https://github.com/rajarshd/Gaussian_LDA

[12] https://github.com/Ardavans/sHDP

[13] We do not compare with the method proposed in (Li et al., 2016b) because its implementation is not available. Moreover the sHDP method, which was published around the same time, is very similar in spirit, but the latter uses a nonparametric HDP topic model.

[14] https://github.com/stanfordnlp/GloVe

[15] https://github.com/facebookresearch/fastText

fastest		india		red		attackers		cession		summer	
Our	GloVe	Our	GloVe	Our	GloVe	Our	GloVe	Our	GloVe	Our	GloVe
slowest	fifth	pakistan	indian	blue	blue	assailants	assailants	ceding	ceding	winter	winter
quickest	second	lanka	mumbai	yellow	white	attacker	besiegers	annexation	ceded	autumn	olympics
slower	sixth	nepal	pakistan	white	yellow	townspeople	pursuers	annexing	reaffirmation	spring	autumn
faster	slowest	indian	pradesh	black	which	insurgents	fortunately	cede	abrogation	year	spring
fast	ever	bangladesh	subcontinent	green	called	policemen	looters	expropriation	stipulating	fall	in
surpassing	quickest	asia	karnataka	pink	bright	retaliation	attacker	continuance	californios	months	beginning
next	third	delhi	bengal	gray	pink	rioters	accomplices	ceded	renegotiation	in	next
surpassed	respectively	sri	bangalore	well	green	terrorists	captors	incorporation	expropriation	also	months
best	tenth	thailand	asia	the	purple	perpetrators	strongpoints	ironically	zapatistas	time	during
slow	first	china	delhi	with	black	whereupon	whereupon	dismantling	annexation	beginning	year

Table 3: Nearest neighbors for selected words.

incisions		unveil		promissory		batgirl		casio	
Our	GloVe	Our	GloVe	Our	GloVe	Our	GloVe	Our	GloVe
incision	incision	unveiling	unveils	issuance	estoppel	catwoman	huntress	notebook	<unk>
indentations	embellishment	utilise	devise	curiously	scribbled	nightwing	zatanna	compute	nightlifepartner
punctures	preferably	introduce	unveiling	wherein	untraceable	supergirl	clayface	practicality	vgnvcm
scalpel	notches	invent	<unk>	handwritten	evidencing	batman	superwoman	utilizing	counterstrike
creases	oftentimes	expose	finalise	ostensibly	gifting	nemesis	gcpd	add	graphing
abrasions	utilising	publicize	solidify	purpotedly	discordant	abandon	supergirl	furthermore	mkii
lacerations	lastly	anticipating	rediscover	omnious	renegotiation	protege	riddler	utilising	kajimitsuo
extractions	silhouettes	unravelling	embellish	phony	repossession	unbeknownst	woman	utilizing	reconditioned
liposuction	discreetly	uncover	reexamine	proposing	waiving	reappears	fight	likewise	bivort
apertures	purposefully	inaugrate	memorializing	ironically	abrogation	cyborg	first	anticipating	spellbinder

Table 4: Nearest neighbors for low-frequency words.

neural		clouds	
Our	GloVe	Our	GloVe
neuronal	neuronal	cloud	cumulonimbus
brain	cortical	shadows	cloud
cortical	correlates	mist	obscured
perceptual	neurons	darkness	mist
physiological	plasticity	heavens	shadows
signaling	neuroplasticity	echoes	aerosols
furthermore	computation	indeed	sky
moreover	circuitry	furthermore	fog
cellular	spiking	fog	swirling
circuitry	mechanisms	lastly	halos

Table 5: Nearest neighbors for high-frequency words.

amazon		apple	
Our	GloVe	Our	GloVe
amazonian	itunes	cherry	iigs
forest	kindle	apples	iphone
brazil	emusic	peach	macintosh
rain	nightlifepartner	pear	itunes
green	astore	red	ipad
trees	cdbaby	sweet	ipod
wildlife	guianas	healthy	ios
preserve	likewise	doctor	microsoft
water	aforementioned	fruit	garbageband
rains	ebay	edible	phone

Table 6: Nearest neighbors for ambiguous words.

were tuned based on a development dataset. To this end, we randomly split our dataset into 60% training, 20% development and 20% testing. We report the results in terms of F1 score on the test set, using the Perf tool[16]. The trained document

[16] http://osmot.cs.cornell.edu/kddcup/software.html

Models	20NG	OHS	TechTC	Reu
TF-IDF	0.852	0.632	**0.306**	0.319
LDA	0.859	0.629	0.305	0.323
HDP	0.862	0.627	0.304	0.339
movMF	0.809	0.610	0.302	0.336
GLDA	0.862	0.629	0.305	0.352
sHDP	0.863	0.631	0.304	0.353
GloVe	0.852	0.629	0.301	0.315
WeMAP	0.855	0.630	**0.306**	0.345
SG	0.853	0.631	0.304	0.341
CBOW	0.823	0.629	0.297	0.339
CvMF	**0.871**	**0.633**	0.305	0.362
CvMF(NIG)	**0.871**	**0.633**	0.305	**0.363**

Table 7: Document classification results (F1).

vectors were used as input to a linear SVM classifier whose trade-off parameter C was tuned from a pool of $\{10, 50, 100\}$, which is a common setting in document classification tasks. Note that our experimental setup is inherently different from those setups where a word embedding model is evaluated on the text classification task using deep neural networks (Jameel et al., 2019), as our focus is on methods that learn document vectors in an unsupervised way. We have therefore adopted a setting where document vectors are used as the input to an SVM classifier.

In our model, we have set the number of word embeddings iterations to 50. The parameters of the vMF mixture model were re-computed after every 5 word embedding iterations. We tuned the dimensionality of the embedding from the pool {100, 150, 200} and the number of vMF mixture components from the pool {200, 500, 800}.

We used the default document topic priors and word topic priors in the LDA and the HDP topic models. For the LDA model, we tuned the number of topics from the pool {50, 80, 100} and the number of iterations of the sampler was set to 1000. We also verified in initial experiments that having a larger number of topics than 100 did not allow for better performance on the development data. The number of vMF mixtures of the comparative method, movMF, was tuned from the pool {200, 500, 800}. For GLDA, as in the original paper, we have used word vectors that were pre-trained using Skipgram on the English Wikipedia. We have tuned the word vectors size and number of topics from a pool of {100, 150, 200} and {50, 80, 100} respectively. The number of iterations of the sampler was again set to 1000. We have used same pre-trained word embeddings for sHDP, where again the number of dimensions was automatically tuned.

Table 7 summarizes our document classification results. It can be seen that our model outperforms all baselines, except for the TechTC dataset, where the results are very close. Among the baselines, sHDP achieves the best performance. Interestingly, this model also uses von Mishes-Fisher mixtures, but relies on a pre-trained word embedding.

Document Retrieval. Next we describe our document retrieval experiments. Specifically, we consider this problem as a learning-to-rank (LTR) task and use standard information retrieval (IR) tools to present our evaluation results.

We have used the following standard IR benchmark datasets: 1) WT2G[17] along with standard relevance assessments and topics (401 - 450), 2) TREC HARD (denoted as HARD)[18], 3) AQUAINT-2 (AQUT)[19] where we considered only the document-level relevance assessments, and 4) LETOR OHSUMED (OHS)[20], which consists of 45 features along with query-document pairs with relevance judgments in five folds. We have obtained the raw documents and queries[21]

[17] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

[18] https://trec.nist.gov/data/hard.html

[19] https://catalog.ldc.upenn.edu/LDC2008T25

[20] https://www.microsoft.com/en-us/download/details.aspx?id=52482

[21] http://ir.dcs.gla.ac.uk/resources/test_collections/

Models	WT2G	HARD	AQUT	OHS
TF-IDF	0.288	0.335	0.419	0.432
LDA	0.291	0.346	0.447	0.461
HDP	0.301	0.333	0.436	0.455
movMF	0.255	0.311	0.421	0.432
GLDA	0.301	0.351	0.447	0.462
sHDP	0.301	0.334	0.437	0.452
GloVe	0.301	0.333	0.436	0.459
WeMAP	0.302	0.362	0.447	0.465
SG	0.301	0.345	0.447	0.461
CBOW	0.299	0.323	0.441	0.459
CvMF	0.305	0.361	0.449	0.469
CvMF(NIG)	**0.306**	**0.363**	**0.450**	**0.471**

Table 8: Document retrieval learning experiments (NDCG@10).

of this dataset, from which we can learn the document representations. As baselines, we have considered the following methods: 1) TF-IDF, 2) LDA (Blei et al., 2003), 3) HDP (Teh et al., 2005), 4) movMF (Banerjee et al., 2005), 5) sHDP (Batmanghelich et al., 2016), 6) GloVe (Pennington et al., 2014), 7) WeMAP (Jameel et al., 2019), 8) Skip-gram, and 9) CBOW word embedding models (Mikolov et al., 2013).

We have adopted the same preprocessing strategy as for the categorization task, with the exception of OHSUMED, for which suitable LTR features are already given. For all other datasets we used the Terrier LTR framework[22] to generate the six standard LTR document features as described in (Jameel et al., 2015). The document vectors were then concatenated with these six features[23]. To perform the actual retrieval experiment, we used RankLib[24] with a listwise RankNet (Burges et al., 2005) model[25]. Our results are reported in terms of NDCG@10, which is a common evaluation metric for this setting.

Our training strategy is mostly the same as for the document categorization experiments, although for some parameters, such as the number of topics and vMF mixture components, we used larger values, which is a reflection of the fact that the collections used in this experiment are substantially larger and tend to be more diverse (Wei and Croft, 2006). In particular, the word vector

[22] http://terrier.org/docs/v4.0/learning.html

[23] Note that in OHS the document vectors were concatenated with 45 LTR features.

[24] https://sourceforge.net/p/lemur/wiki/RankLib/

[25] Note that in principle any LTR model for IR could be used.

Models	20NG	OHS	TechTC	Reu
TF-IDF	0.323	0.288	0.391	0.209
LDA	0.453	0.355	**0.455**	0.221
HDP	0.444	0.321	0.451	0.221
movMF	0.331	0.223	0.422	0.212
GLDA	0.466	**0.356**	**0.455**	0.234
sHDP	0.453	**0.356**	**0.455**	0.236
GloVe	0.455	0.352	0.453	0.221
WeMAP	0.456	0.354	0.454	0.223
SG	0.453	0.355	0.453	0.221
CBOW	0.432	0.344	0.421	0.220
CvMF	**0.492**	**0.356**	**0.455**	**0.239**
CvMF(NIG)	**0.492**	**0.356**	**0.455**	0.236

Table 9: Word coherence results in c_v computed using Gensim.

lengths were chosen from a pool of {150, 200, 300} and the vMF mixtures from a pool of {300, 1000, 3000}. In the LDA model, we selected the number of topics from a pool of {100, 150, 200}. For GLDA we have used the same pool for the number of topics. All our results are reported for five-fold cross validation, where the parameters of the LTR model were automatically tuned, which is a common LTR experimental setting (Liu et al., 2015a).

The results are presented in Table 8, showing that our model is able to consistently outperform all methods. Among the baselines, our NIG variant achieves the best performance in this case, which is remarkable as this is also a word embedding model.

Word Coherence. In traditional topic models such as LDA, the topics are typically labelled by the k words that have the probability in the topic. These words tend to reflect semantically coherent themes, which is an important reason for the popularity of topic models. Accordingly, measuring the coherence of the top-k words that are identified by a given topic model, for each topic, is a common evaluation measure (Shi et al., 2017). Using the configurations that performed best on the tuning data in the document categorization task above, we used Gensim[26] (Řehůřek and Sojka, 2010) to compute the coherence of the top-20 words using the c_v metric (Röder et al., 2015). For our model, GDLA and sHDP, the mixture components that were learned were consided as topics for this experiment. For GloVe, WeMAP, SG, TF-IDF, and CBOW, we used the von Mises-Fisher (vMF) soft clustering model (Banerjee et al., 2005) to determine the cluster memberships of the context words. For the TF-IDF results, we instead used hard vMF clustering (Hornik and Grün, 2014), as the movMF results are based on TF-IDF features as well. We tuned the number of clusters using the tuning data. The top-20 words after applying the clustering model were then output based on the distance from the cluster centroid.

[26] radimrehurek.com/gensim/models/coherencemodel.html

The results are shown in Table 9, showing that the word clusters defined by our mixture components are more semantically coherent than the topics obtained by the other methods.

5 Conclusions

In this paper, we analyzed the effect of adding a prior to the GloVe word embedding model, encoding the intuition that words can be organized in various natural groupings. Somewhat surprisingly, perhaps, this leads to a word embedding model which behaves substantially differently from existing methods. Most notably, our model substantially outperforms standard word embedding models in analogy tasks that focus on syntactic/morphological relations, although this comes at the cost of lower performance in semantically oriented tasks such as measuring word similarity. We also found that the model performs better than standard word embedding models when it comes to modelling rare words.

Word embedding models can also be used to learn document embeddings, by replacing word-word co-occurrences by document-word co-occurrences. This allowed us to compare our model with existing approaches that use von Mises-Fisher distributions for document modelling. In contrast to our method, these models are based on topic models (e.g. they typically model documents as a multinomial distribution over topics). Surprisingly, we found that the document representations learned by our model outperform these topic modelling-based approaches, even those that rely on pre-trained word embeddings and thus have an added advantage, considering that our model in this setting is only learned from the (often relatively small) given document collection. This finding puts into question the value of document-level topic distributions, which are used by many document embedding methods (being inspired by topic models such as LDA).

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382.

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *Proceedings ACL*, volume 2016, pages 537–542.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of machine Learning research*, 3:993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings ACL*, pages 795–804.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, pages 1606–1615.

S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings ACL*, pages 84–94.

Junxian He, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P Xing. 2017. Efficient correlated topic modeling with topic embedding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 225–233.

Kurt Hornik and Bettina Grün. 2014. movMF: An R package for fitting mixtures of von mises-fisher distributions. *Journal of Statistical Software*, 58(10):1–31.

Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P. Xing. 2015. Entity hierarchy embedding. In *ACL*, pages 1292–1300.

Shoaib Jameel, Zihao Fu, Bei Shi, Wai Lam, and Steven Schockaert. 2019. Word embedding as maximum a posteriori estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Shoaib Jameel, Wai Lam, and Lidong Bing. 2015. Supervised topic models with word order structure for document classification and retrieval learning. *Information Retrieval Journal*, 18(4):283–330.

Shoaib Jameel and Steven Schockaert. 2016. Entity embeddings with conceptual subspaces as a basis for plausible reasoning. In *Proceedings of ECAI*, pages 1353–1361.

Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016a. Generative topic embedding: a continuous representation of documents. In *Proceedings ACL*.

Ximing Li, Jinjin Chi, Changchun Li, Jihong Ouyang, and Bo Fu. 2016b. Integrating topic modeling with word embeddings by mixtures of vmfs. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 151–160.

Yuezhang Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia P. Sycara. 2016c. Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In *Proceedings COLING*, pages 2678–2688.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015a. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of ACL*, pages 1501–1511.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015b. Topical word embeddings. In *Proceedings AAAI*, pages 2418–2424.

Kanti V Mardia and Peter E Jupp. 2009. *Directional statistics*, volume 494. John Wiley & Sons.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings NAACL-HLT*, pages 142–148.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings ICLR*.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Mohammad Taher Pilehvar, Dimitri Kartsaklis, Victor Prokhorov, and Nigel Collier. 2018. Card-660: Cambridge rare word dataset-a reliable benchmark for infrequent word representation models. *arXiv preprint arXiv:1808.09308*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

Bei Shi, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. Jointly learning word embeddings and latent topics. In *Proceedings SIGIR*, pages 375–384.

Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

Xing Wei and W Bruce Croft. 2006. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM.

C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T.-Y. Liu. 2014. RC-NET: A general framework for incorporating knowledge into word representations. In *Proc. CIKM*, pages 1219–1228.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings ACL*, pages 545–550.