Personalized Semantic Word Vectors

Javid Ebrahimi
University of Oregon
Eugene, Oregon 97403, USA
javid@cs.uoregon.edu

Dejing Dou
University of Oregon
Eugene, Oregon 97403, USA
dou@cs.uoregon.edu

ABSTRACT

Distributed word representations are able to capture syntactic and semantic regularities in text. In this paper, we present a word representation scheme that incorporates authorship information. While maintaining similarity among related words in the induced distributed space, our word vectors can be effectively used for some text classification tasks too. We build on a log-bilinear document model (lbDm), which extracts document features, and word vectors based on word co-occurrence counts. First, we propose a log-bilinear author model (lbAm), which contains an additional author matrix. We show that by directly learning author feature vectors, as opposed to document vectors, we can learn better word representations for the authorship attribution task. Furthermore, authorship information has been found to be useful for sentiment classification. We enrich the author model with a sentiment tensor, and demonstrate the effectiveness of this hybrid model (lbHm) through our experiments on a movie review-classification dataset.

Keywords

word vectors; document model; author model

1. INTRODUCTION

Distributed word representations are now ubiquitous in the NLP community. Neural models [3] map words to an n-dimensional space through language modeling. This allows learning multiple dimensions of similarity that encode the syntactic, semantic [12] features of words in a compressed, numerical format. These models generally do not utilize the statistics of the document, nor the corpus, because they are trained on local context windows [17]. However, the learned vectors can be fine-tuned by discriminative neural networks and can be used for text classification tasks [20, 8]. Perhaps Le and Mikolov's Paragraph Vector [9] can be distinguished from others since they have extended the `word2vec` [12] to allow for word, sentence and document vectors in a unified framework.

Log-bilinear document model (lbDm) [11] uses term-document data to learn semantic word vectors. It derives a probabilistic model with a log-bilinear energy function to model the document, as opposed to sequences of word windows. The lbDm uses the bag-of-words assumption and its formulation is similar to LDA [4]. However, it defines a novel energy function and the learned document features are not confined to the topic simplex. Despite the simplicity of the model, it works well in text classification tasks. It has been applied to sentiment classification [10], document classification [7], and information retrieval [23].

We extend lbDm in two ways. First, we show that by sharing the document features, based on authorship, we can find personalized semantic word vectors that perform significantly better in the authorship attribution task. Second, combined with sentiment matrices, these new author-sentiment-biased word vectors can improve the results of sentiment classification. In our hybrid model, the negative energy of a word, given the document, its author, and its sentiment, is proportional to the dot product of the word vector and its sentiment-transformed author vector. Our idea in employing authorship information could be applied to other document or word embedding models as well.

2. RELATED WORK

This work is based on a log-bilinear language model proposed in [13, 14]. They used Restricted Boltzmann Machines for language modeling. That model was later modified by [11, 10] to develop a log-bilinear document model (lbDm) for sentiment classification and subjectivity recognition. The lbDm is based on the bag-of-words assumption, and the probability of a word, given the document features, can be defined by a softmax function. Introducing the document feature vector helped their model to go beyond a window of words, to learn word representations in the context of the whole document.

Our work is also related to personalization, which has been investigated in different tasks, including language modeling [26, 6], author topic modeling [18], authorship attribution [19], and sentiment classification [1, 21]. For instance, Wachsmuth et al. [25] use an argumentation model for each reviewer to classify the sentiment in the movies. In [15] an author-specific generative model of a movie review is employed, in order to take author preferences and writing style into account, so as to improve sentiment classification. Results in [21] show improved word embedding after taking the user and the product evidence into account. Similarly, Hovy [5] shows that sentiment analysis can be improved by the inclusion of author demographics, such as age and gender.

3. LOG-BILINEAR AUTHOR MODEL

In the log-bilinear Document model (lbDm) [11], a document is represented using a continuous mixture distribution over words indexed by a document-specific random variable θ. Based on the

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983875

bag-of-words assumption, words in a document are conditionally independent given the independent mixture variable θ. Thus, the probability of a document is,

$$p(d) = \int p(d, \theta) d\theta = \int p(\theta) \prod_{i=1}^{N} p(w_i|\theta) d\theta$$

where N is the number of words in d and w_i is the i_{th} word in d.

In the log-bilinear author model (lbAm), we have an author representation matrix $U \in \mathbb{R}^{\beta \times |A|}$ where each author a (represented as a one-hot vector) in the author set A, has a β dimensional vector representation $Ua = \psi_a$, corresponding to that author's column in U. Thus, the probability of a document given its author is,

$$p(d, a) = \int p(d, \psi_a) d\psi = \int p(\psi) \prod_{i=1}^{N} p(w_i|\psi, a) d\psi \qquad (1)$$

We define the following energy function,

$$E(w; \psi, \phi, a, b) = -\psi_a^T \phi_w - b_w$$

The energy function uses a word embedding matrix $R \in \mathbb{R}^{\beta \times |V|}$, where each word w (represented as a one-hot vector) in the vocabulary V, has a β dimensional vector representation $\phi_w = Rw$, corresponding to that word's column in R. A bias b_w for each word to capture differences in overall word frequencies is also added. For the distribution $p(w|a)$, we use a standard softmax function,

$$p(w|a; U, R, b) = \frac{\exp(\psi_a^T \phi_w + b_w)}{\sum_{w' \in V} \exp(\psi_a^T \phi_{w'} + b_{w'})}$$

Unlike lbDm, we do not have any document-specific features and are only interested in authors' vectors. This model is more useful in applications such as authorship attribution or author topic modeling, which address problems such as: *Who are likely to have written documents similar to an observed document?* and *Which authors produce similar papers?* [18].

4. LOG-BILINEAR HYBRID MODEL

Authorship information can also improve sentiment classification. Different people use different diction to describe similar views. Thus, a sentiment classifier that uses authorship can perform better. Hybrid model (lbHm) aims to take both authorship and sentiment information into account. One way to do this is to use linear additions of sentiment-shared and authorship-shared vectors. This might result in problems of scaling and of tuning. Instead, we represent sentiments by a tensor γ, and use the following energy function,

$$E(w; \psi, \gamma, \phi, a, s, b) = -\psi_a^T \gamma^{[s]} \phi_w - b_w$$

where $\gamma^{[s]} \in \mathbb{R}^{\beta \times \beta}$ is a slice in the sentiment tensor that denotes the transformation matrix associated with the sentiment s. This representation realizes more complex interactions between sentiment and authorship. In addition, the transformation of the author vector, based on sentiment, helps capture the features of the sentiment-bearing words that a specific author uses.

5. LEARNING

Here, we discuss the learning algorithm for lbHm, which is more general than lbAm. Given an i.i.d document collection D, let d_k denote the k_{th} document, with author a, and sentiment s. We aim to learn model parameters R and b. Using MAP estimates for ψ and γ, we can approximate this learning problem as,

$$L(D) = \prod_{d_k \in D} p(\hat{\psi_a}) p(\hat{\gamma}) \prod_{i=1}^{N_k} p(w_i|\hat{\psi_a}, \hat{\gamma}, s; R, b)$$

where $\hat{\psi_a}$ denotes the MAP estimate of author vector ψ_a, and $\hat{\gamma}$ is the MAP estimate of the sentiment tensor γ. Adding the regularization terms, the final log-likelihood learning problem is,

$$\max_{R,b} \lambda \parallel R \parallel_F + \lambda \parallel \hat{\gamma} \parallel_F + \lambda \parallel U \parallel_F + \sum_{d_k \in D} \sum_{i=1}^{N_k} \log p(w_i|\hat{\psi_a}, \hat{\gamma}, s; R, b) \qquad (2)$$

To optimize this non-convex objective function, we use coordinate ascent. First, we optimize the word representations (R and b) while leaving the MAP estimates $\hat{\psi_a}$ and $\hat{\gamma}$ fixed. Then we find $\hat{\psi_a}$ while leaving the R, b, and $\hat{\gamma}$ fixed, and finally, we do the same for $\hat{\gamma}$. We continue this process until no further improvement is gained.

The contributions made by a document d_k from D to the derivatives of $L(D)$ with regard to the slice s of the tensor γ are given by,

$$\sum_{i=1}^{N_k} \frac{\partial \log p(w_i|\hat{\psi_a}, \hat{\gamma}, s)}{\partial \gamma^{[s]}} = N_k \Big\langle \hat{\psi_a} \phi_{w_i}^T \Big\rangle_D - N_k \Big\langle \hat{\psi_a} \phi_{w_i}^T \Big\rangle_M \qquad (3)$$

where N_k is the length of the document and $\langle . \rangle_D$ and $\langle . \rangle_M$ denote expectations w.r.t. word count distribution and $p(w_i|\hat{\psi_a}, \hat{\gamma}, s)$, respectively. Partial derivatives with regard to $\hat{\psi_a}$, R and b can be computed similarly. Computation of these derivatives can be performed linearly, in the size of the vocabulary, and we use LBFGS for optimization.

6. EXPERIMENTS

In this section we discuss our experiments, in which we use two datasets for sentiment classification [16] and authorship attribution [2]. We make comparisons with some other word representation schemes. After learning the word embeddings, we use the mean representation vector as the document features, i.e., the average vector of all the words present in the document. For all the methods in the experiments, we used SVM trained with both polynomial and linear kernels and report the best of the two. In both experiments, we could add extra features to improve the accuracy and compete with state-of-the-art results. The focus of our work, however is to show the superiority of the word representation model compared with other schemes.

We set the regularization parameter λ to 10^{-4} and the number of the dimensions of the word vectors β to 50. Following [24], we control the standard deviation of the embeddings, and set the scaling hyper-parameter to 0.1.

6.1 Authorship Attribution

Authorship attribution deals with identifying the authors of documents. We follow the setup of the PAN'11 competition (Argamon and Juola, 2011): We train the models on the training subset, tune the parameters according to the given validation subset, and run the tuned models on the given testing subset. The training dataset consists of 9337 documents and 72 authors, while the validation and test sets contain 1298 and 1300 documents respectively.

For our baselines, we use the word topic distributions from the Author Topic model (AT) [18], bag-of-words, lbDm [11], and continuous bag-of-words (CBOW) [12]. We also compare our results with the Paragraph Vector model (PV) [9], which can be used for joint word representation learning and supervised learning.

For classification purposes in PV, the paragraph vector for instances with similar labels can be shared. We used the `gensim` implementation [1] of PV.

Method	Test	Validation
lbDm	25.64	27.48
AT	30.50	33.15
BOW	38.43	39.68
CBOW	13.89	14.66
PV	33.38	32.04
lbAm	**41.41**	**42.46**

Table 1: Accuracy on PAN'11 Dataset

The poor performance of lbDm can be attributed to the large number of classes and the inability of the model to differentiate among authors' interests by treating every document independently of the others. Moreover, compared with AT, distributed representation of lbAm is not confined to the topic simplex, and it can better capture semantic similarities. While the representations produced by CBOW have high quality (Table 2), they perform poorly on the classification task. Similarly, PV yields a low accuracy on authorship attribution. The reason for this is that PV would perform better when it has access to a large amount of labeled/unlabeled data. This becomes even more challenging for PV in a problem with many classes (72 in this dataset).

A qualitative assessment, based on cosine similarity of the words, and some query words can be seen in Table 2. We compare lbAm with AT, and CBOW using the `word2vec` tool[2].

	lbAm	AT	CBOW
company	person	forward	owned
	send	notice	recently
	business	asap	industry
	products	referencing	consultant
	texas	spend	marketplace
counsel	affiliates	havoc	general
	locate	proposed	president
	courts	portions	vice
	merged	courts	litigation
	attorneys	affadavit	attorney
agreements	discussions	purpose	netting
	agreement	replacing	master
	assist	portions	employment
	standard	trail	brokerage
	provisions	finalized	confidentiality
parties	version	conform	perform
	agreed	exelon	states
	prices	llc	attorneys
	discussed	legal	provisions
	benefit	locational	context

Table 2: Similarity of learned word vectors by lbAm, AT, and CBOW. Each query word is listed with its 5 most similar words, based on the cosine similarity of the vectors.

6.2 Sentiment Classification

To evaluate our hybrid model (lbHm), we use the well-known movie review dataset [16], which contains 2000 reviews and 312 authors. In order to have sufficient data per author, we created two datasets, wherein only the authors with at least five (dataset-5), and at least ten (dataset-10) reviews of each polarity are considered. This reduces the number of reviews in the first dataset to 1239 (552 positive, and 687 negative) with 48 authors, and the number in the second dataset to 788 (332 positive and 456 negative) with 25 authors. Results of 10-fold cross validation are reported in Table 3.

Method	dataset-5	dataset-10
LDA	74.98	75.10
lbDm	81.09	80.83
BOW	87.49	85.40
CBOW	79.82	80.58
PV	83.53	81.77
lbSm	86.70	85.94
lbHm	**88.05**	**86.59**

Table 3: Accuracy of sentiment classification

We compare our results with LDA [4], lbDm, CBOW, PV, and bag-of-words. The lbSm is similar to lbAm, where the document features are shared based on the sentiment, instead of authorship. Our lbHm takes both sentiment and authorship into account, and it achieves the best results.

It is interesting to see the relatively poor performance of PV on these datasets. It had achieved impressive performance on another movie review dataset [10], in a transductive setting, with a large number of labeled and unlabeled training instances. Whereas here, we have a relatively small dataset with no unlabeled training instances. As another contributing factor which was also demonstrated in [22], the average length of the reviews in that dataset (i.e., 227) is a almost a third of the average length of the dataset that we used (i.e., 646)

As a qualitative evaluation of the variables in the lbHm model, we visualize the author vector, before (ψ_a), and after applying the positive and negative sentiment matrices ($\psi_a^T \gamma^{[s]}$) in Table 4. Authors are represented by the words that have the highest cosine similarity with the author vectors.

The words in bold font are the sentiment-bearing words that rise up, among the top words, as most similar to the author vector, in the sentiment transformed space. The words that are underlined are sentiment-bearing words that are among the words most similar to the original author vector, which might or might not remain among the top-ranked words after the transformation. IbHm captures relationships between authorship and sentiment variables, and it produces word vectors that are also suitable for a sentiment classification task.

7. CONCLUSIONS

In this paper, we presented two extensions of a previous log-bilinear document model (lbDm) to induce semantic word representations. This bilinear model can be efficiently modified, using parameter sharing or linear operations for improved document classification. Log-bilinear author model (lbAm) learns personalized word representations by sharing the document features based on authorship. Knowing the author of a document can also improve sentiment classification. Log-bilinear hybrid model (lbHm), in addition to author vectors, uses a sentiment tensor to enrich the model based on the sentiment of the document. The advantage of our models is that while they produce high-quality word vectors, they also can be readily fed to standard classifiers.

A natural extension to this work is to apply similar formulations to other document modeling techniques. For example, the Para-

[1] http://radimrehurek.com/gensim/

[2] https://code.google.com/archive/p/word2vec/

	auth	auth-pos	auth-neg
Reviewer-12	killing	killing	mixture
	mixture	**transcend**	cringed
	cringed	mixture	hairstylist
	personal	pleasant	stinker
	galoshes	hairstylist	distract
	community	sex-driven	believable
	pleasant	terrifying	stalwart
	hairstylist	ferris	**unscrupulous**
	hot	thoroughly	railway
	terrifying	confrontations	recognizable
	ferris	recognizable	non-cynics
	stinker	psychopath	ultra-low
	sharing	**competent**	faucet
	distract	arabs	**ludicrous**
Reviewer-16	drives	pleasant	tracy
	parody	sex-driven	screenwriters
	vancouver	simple	deeds
	pleasant	merely	sitcom
	number	performance	**decrepit**
	sex-driven	**intricately**	kravitz
	terrifying	molly	carving
	london's	**scrumptious**	**vicious**
	late	cuteness	attacks
	june	story	publique
	merely	play's	performance
	mushy	herb	unique
	dinosaurs	**moralistic**	**horribly**
	attacks	surroundings	realistic

Table 4: The author vector, for two reviewers in the movie-review polarity dataset, before (ψ_a), and after applying the positive and negative sentiment matrices ($\psi_a^T \gamma^{[s]}$). In all columns, words that have the highest cosine similarity with the vector are shown. The bold and underlined words bear sentiment.

graph Vector [9], or the Predictive Text Embedding [22] could also be extended to allow for a combination of authorship and sentiment features. In addition, employing external features sets for the tasks could improve the results presented in this paper.

8. ACKNOWLEDGMENT

This work was partially supported by the NIH grant R01GM103309.

9. REFERENCES

[1] M. Al Boni, K. Zhou, H. Wang, and M. S. Gerber. Model adaptation for personalized opinion analysis. In *Proceedings of ACL-IJCNLP*, pages 769–774, 2015.

[2] S. Argamon and P. Juola. Overview of the international authorship identification competition at PAN-2011. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] D. Hovy. Demographic factors improve classification performance. In *Proceedings of ACL-IJCNLP*, pages 752–762, 2015.

[6] Y.-Y. Huang, R. Yan, T.-T. Kuo, and S.-D. Lin. Enriching cold start personalized language model using social network information. In *Proceedings of ACL*, pages 611–617, 2014.

[7] H. Jing, Y. Tsao, K.-Y. Chen, and H.-M. Wang. Semantic naïve Bayes classifier for document classification. In *Proceedings of the IJCNLP*, pages 1117–1123, 2013.

[8] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665, 2014.

[9] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In T. Jebara and E. P. Xing, editors, *Proceedings of ICML*, pages 1188–1196, 2014.

[10] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, pages 142–150, 2011.

[11] A. L. Maas and A. Y. Ng. A probabilistic model for semantic word vectors. In *Workshop on Deep Learning and Unsupervised Feature Learning, NIPS*, 2010.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119. 2013.

[13] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of ICML*, pages 641–648, 2007.

[14] A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of ICML*, pages 1751–1758, 2012.

[15] S. Mukherjee, G. Basu, and S. Joshi. Joint author sentiment topic model. In *Proceedings of SDM*, pages 370–378, 2014.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[17] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.

[18] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of UAI*, pages 487–494, 2004.

[19] Y. Seroussi, F. Bohnert, and I. Zukerman. Authorship attribution with author-aware topic models. In *Proceedings of ACL*, pages 264–269, 2012.

[20] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642, 2013.

[21] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of ACL-IJCNLP*, pages 1014–1023, 2015.

[22] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of SIGKDD*, pages 1165–1174, 2015.

[23] X. Tu, J. Luo, B. Li, and T. He. Log-bilinear document language model for ad-hoc information retrieval. In *Proceedings of CIKM*, pages 1895–1898, 2014.

[24] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394, 2010.

[25] H. Wachsmuth, M. Trenkmann, B. Stein, and G. Engels. Modeling review argumentation for robust sentiment analysis. In *Proceedings of COLING*, pages 553–564, 2014.

[26] G.-R. Xue, J. Han, Y. Yu, and Q. Yang. User language model for collaborative personalized search. *ACM Transactions on Information Systems*, 27(2):11:1–11:28, 2009.