A Probabilistic Approach to Sense Embeddings

Anuj Nagpal
14116

Divyat Mahajan
14227

Rushab Munot
14405

1 Problem Description

To model any language, we must have a way to represent its words. One way coudl be to represent every word with a one-hot vector corresponding to its dictionary position but that won't contain any useful semantic information, as in distances between word vectors will only denote the difference in their alphabetic ordering. A better approach will be where words with similar meanings represent nearby points in vector space ([1] Mikolov Tomas, et al., 2013)

[2] Vilnis and McCallum proposed one method which represented words by a whole Gaussian distribution instead of a single one point vector and the model learnt its mean and covariance matrix. The idea of point embedding was captured by the mean vector of the Gaussian distribution along with the extra useful information like probability mass and uncertainity across a set of semantics provided by the full distribution.

However, a one vector per word approach is inadequate to model polysemous words. A single word can have multiple meanings when used in different contexts. Consider two senses of the word bank (which has many more senses) - one pertaining to the financial sense, and the other to the bank of a river. These two senses of bank are hardly related to each other in any way, however both of them have the same vector, which is sort of a weighted combination of the two senses.

Since a Gaussian distribution can have only one mode, the learned uncertainty for a polysemous word (words with multiple distinct meanings), can be overly diffuse in order for the model to assign some density to any plausible semantics. Moreover, the mean of the Gaussian can be pulled in many opposing directions, leading to a biased distribution that centers its mass mostly around one meaning while leaving the others not well represented. Thus, we need a better model that can learn multiple vectors per word, ever vector corresponding to the sense of a word.

2 Literature Survey

[3] Ben Athiwaratkun, Andrew Gordon Wilson. Multimodal Word Distributions

Athiwaratkun and Wilson proposed a probabilistic word embedding that can capture multiple meanings. They modeled each word with a mixture of Gaussians and learnt the parameters using a maximum margin energy-based rankin objective where the energy function describes the affinity between a pair of words.

Word Representation

They represented each word w as a Gaussian mixture with K components such that the density function of w is given by:

$$p_w(\vec{x}) = \sum_{i=1}^{K} p_{w,i}\mathcal{N}[\vec{x}; \vec{\mu}_{w,i}, \Sigma_{w,i}]$$

$$= \sum_{i=1}^{K} \frac{p_{w,i}}{\sqrt{2\pi|\Sigma_{w,i}|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_{w,i})^T \Sigma_{w,i}^{-1}(\vec{x}-\vec{\mu}_{w,i})}$$

where $\sum_{i=1}^{K} p_{w,i} = 1$.

So the model has three parameters - $p_{w,i}$ representing the component probability (mixture weight), $\vec{\mu}_{w,i}$ representing the location of the i_{th} component of word w (similar to point embeddings) and $\Sigma_{w,i}$, covariance matrix of i_{th} Gaussian and contains the uncertainty information. Learning these three parameters draws inspiration from the continuous skip-gram model ([1] Mikolov et al., 2013a), where word embeddings are trained to maximize the probability of observing a word given another nearby word.

2.1 Energy Function

Consider two pairs of words (w, c) and (w, c') where w is sampled from a sentence in a corpus and c is a nearby word within a context window of length (say l) and c is a negative context word. For example, a word w = 'rock' which occurs in the sentence 'I listen to rock music' has context words (I, listen, to, music) and c' is a negative context word obtained from random sampling (e.g. tortoise). The idea behind the energy function is to maximize the energy between words that occur near each other (w and c) and minimize the energy between w and c'.

Athiwaratkun and Wilson used a max-margin ranking objective used for Gaussian embeddings which pushes the similarity of a word and its positive context higher than that of its negative context by a margin m:

$$L_\theta(w, c, c') = max(0, m - logE_\theta(w, c) + logE_\theta(w, c'))$$

where for gaussian mixtures f and g representing words w_f and w_g, log-energy is:

$$logE_\theta(f, g) = \int \left(\sum_{i=1}^{K} p_i\mathcal{N}(x; \vec{\mu}_{f,i}, \Sigma_{f,i})\right) \times \left(\sum_{i=1}^{K} q_i\mathcal{N}(x; \vec{\mu}_{g,i}, \Sigma_{g,i})\right) dx$$

$$= log\sum_{j=1}^{K}\sum_{i=1}^{K} p_i q_j e^{\xi_{i,j}}$$

where

$$\xi_{i,j} \equiv log\mathcal{N}(0; \vec{\mu}_{f,i} - \vec{\mu}_{g,j}, \Sigma_{f,i} + \Sigma_{g,j})$$

$$= -\frac{1}{2}log(det(\Sigma_{f,i} + \Sigma_{g,j})) - \frac{D}{2}log(2\pi) - \frac{1}{2}(\vec{\mu}_{f,i} - \vec{\mu}_{g,j})^T \Sigma_{w,i}^{-1}(\vec{\mu}_{f,i} - \vec{\mu}_{g,j})$$

They minimized this objective by mini-batch stochastic gradient descent with respect to mean vectors ($\vec{\mu_{w,i}}$), covariance matrices ($\Sigma_{w,i}$) and mixture weights ($p_{w,i}$) as parameters.

[2] **Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space.**

Neelakantan et. al. propose a method based on context clustering for sense vector generation. They maintain a global vector for each word as well as multiple sense vectors. The context is embedded as a sum of the global vectors of the words in the context (which is not a very good way to model clusters). They cluster these average contexts, increasing the number of clusters every time a dissimilar con- text (i.e. no similar to the earlier clutsers) is observed, thus learning the number of senses per word (a non parametric approach). This is essentially a Word Sense Disambiguation layer, which is added before the skipgram layer in wor2vec. Thus a disambiguated sense vector is used to predict the context in the skip gram layer.

3 Approaches

In the model proposed by Athiwaratkun and Wilson [1], the number of parameters to be trained are quite large: $KND + KND^2 + NK$ and their model could train values assuming 2 components in Gaussian mixture model or two senses per word (which might not be sufficient for all words). So, one thought that comes in mind that reducing the number of parameters to be trained could help in increasing the value of K (number of components in Gaussian mixture).
So we modified their model with the objective of reducing parameters by trying two approaches:

3.1 Approach 1

We propose an approach where instead of learning a separate Gaussian for each sense of a word, we learn a set of basis Gaussian vectors ($\{(\mu_1, \Sigma_1), (\mu_2, \Sigma_2), \ldots, (\mu_m, \Sigma_m)\}$
Using this set of basis Gaussian vectors, we now model the k_{th} sense of word w by:

$$w_{i.k} = \sum_{m=0}^{M} z_{i,k,m}\mathcal{N}(\mu_m, \Sigma_m) \tag{1}$$

where $z_{i,k,m}$ serve as latent variables for each w to be learned.

So, for the i^{th} word, w_i, we incorporate all the senses as

$$w_i = \sum_{k=1}^{K} \pi_{i,k}(\sum_{m=1}^{M} z_{i,k,m}\mathcal{N}(\mu_m, \Sigma_m)) \tag{2}$$

Now if $\psi_{i,m} = \sum_{k=1}^{K} \pi_{i,k} z_{i,k,m}$, the equation becomes

$$w_i = \sum_{m=1}^{M} \psi_{i,m}\mathcal{N}(\mu_m, \Sigma_m) \tag{3}$$

$$\Psi_i = Z_i \times \Pi_i \tag{4}$$

This model has total parameter to be learned as $MD + MD^2 + NKM + NK$ as compared to $KND + KND^2 + NK$ parameters in previous case. Taking $N = 10^5$, $K = 5$, M approximately equal to D ($M = 100$ and $D = 150$), we observe roughly 100 times reduction in number of parameters to be learnt. Note that we do not make any change to energy based loss function and gradient optimization described in Athiwaratkun and Wilson [3].
Thus the word vector for every word can be effectively written as a linear mixture of the original basis Gaussians (as compared to a mixture over the distributions of its k senses). This can help in training the model easily as now we only estimate $\{\psi_1, \ldots, \psi_m\}$ instead of $\{\pi_1, \ldots, \pi_k\}$ and $z_{k,m}$ for $k = 1, 2, \ldots, K$ and $m = 1, 2, \ldots, M$.
But learning only the $\psi_i's$ generates sense vectors and not the word vectors. To generate vectors for every sense we need $\Pi_i's$ and $Z_i's$.
Hence, we need to determine a way to learn Π_i and Z_i given Ψ_i. The relation between them for a particular word w is given by equation (4) with below constraints

$$\sum_{k=1}^{K} \pi_{i,k} = 1 \text{ and } \sum_{m=1}^{M} z_{i,k,m} = 1 \Rightarrow \sum_{m=1}^{M} \psi_{i,m} = 1$$

To solve this problem, we need to alternately optimize between Π_i and Z_i for given Ψ_i. We also need to describe a suitable loss function for this task. However, multiple solutions exist for the relationship described in equation (2). To actually compare senses across words there must be a specific relation between Π_i and Z_i for all words w_i. But this will not be the case with Iterative Gradient Based Optimization methods. So we move on to a second approach.

3.2 Approach 2

What if we follow the same model as done by Athiwaratkun and Wilson but express each $\mu_{w,k}$ and $\Sigma_{w,k}$ as:

$$\mu_{w,k} = \sum_{m=1}^{M} z_{w,k,m}\mu_m \tag{5}$$

$$\Sigma_{w,k} = \sum_{m=1}^{M} z_{w,k,m}\Sigma_m \tag{6}$$

where μ_m and Σ_m are shared by all words. This makes the equation for i_{th} word look like:

$$w_i = \sum_{k=1}^{K} \pi_{w,k}\mathcal{N}(\sum_{m=1}^{M} z_{i,k,m}\mu_m, \sum_{m=1}^{M} z_{i,k,m}\Sigma_m) \tag{7}$$

where $z_{i,k,m}$ will serve as latent variables for each w to be learned. The number of parameters in this case will be $NMK + NK + MD + MD^2$, which is again less than the original number of parameters.

4 Experimentation and Results

We modified the code used by Athiwaratkun and Wilson and modified it as per our approaches.

Metric Used

Lexical entailment between words is denoted by $w_1 \models w_2$ which means that all instances of w_1 are w_2 . For example, $aircraft \models vehicle$ will be a positive pair and $aircraft \models insect$ will be a negative score. We generate entailment scores of word pairs and find the best threshold, measured by Average Precision (AP) or F1 score, which identifies negative versus positive entailment [3].

Comparative Analysis

Following are the scores obtained on various approaches:
Note: All these scores are reported with spherial covariance matrices.

Approach	K	M	D	Epochs Trained	Average Precision	F1 Score
Original	2	N.A.	50	10	67.72	72.44
Approach 1	2	10	50	4	50.00	57.00
Approach 2	2	300	50	20	63.43	68.34
Approach 2	2	500	50	14	62.63	68.09
Approach 2	4	500	50	5	57.65	66.68

Comparison for various datasets (the values for approach 2 are corresponding to M=300 and K=2)

Dataset	Approach 2 Score	Original Score
SL	18.82	20.34
WS	38.10	56.06
WS-S	43.19	61.45
WS-R	34.53	52.47
MEN	41.84	58.94
MC	25.36	50.68
RG	33.03	50.24
YP	24.39	27.89
MT-287	43.46	60.01
MT-771	39.01	51.31
RW	6.84	11.71
SCWS_maxdot	38.25	43.53
AVERAGE	31.69	45.55

Word	Context	Nearest Neighbors
Bank	0	’bank:0’, ’banking:0’, ’banks:0’, ’financial:1’, ’privatization:0’, ’monetary:1’, ’investment:1’, ’finance:0’, ’loans:0’, ’investments:0
Bank	1	’bank:1’, ’rabat:0’, ’dubai:1’, ’karachi:0’, ’hamlets:0’, ’gaza:1’, ’strip:1’, ’cccc:0’, ’kuala:1’, ’jordan:1

Table 1: Highest Similarity Words for Bank

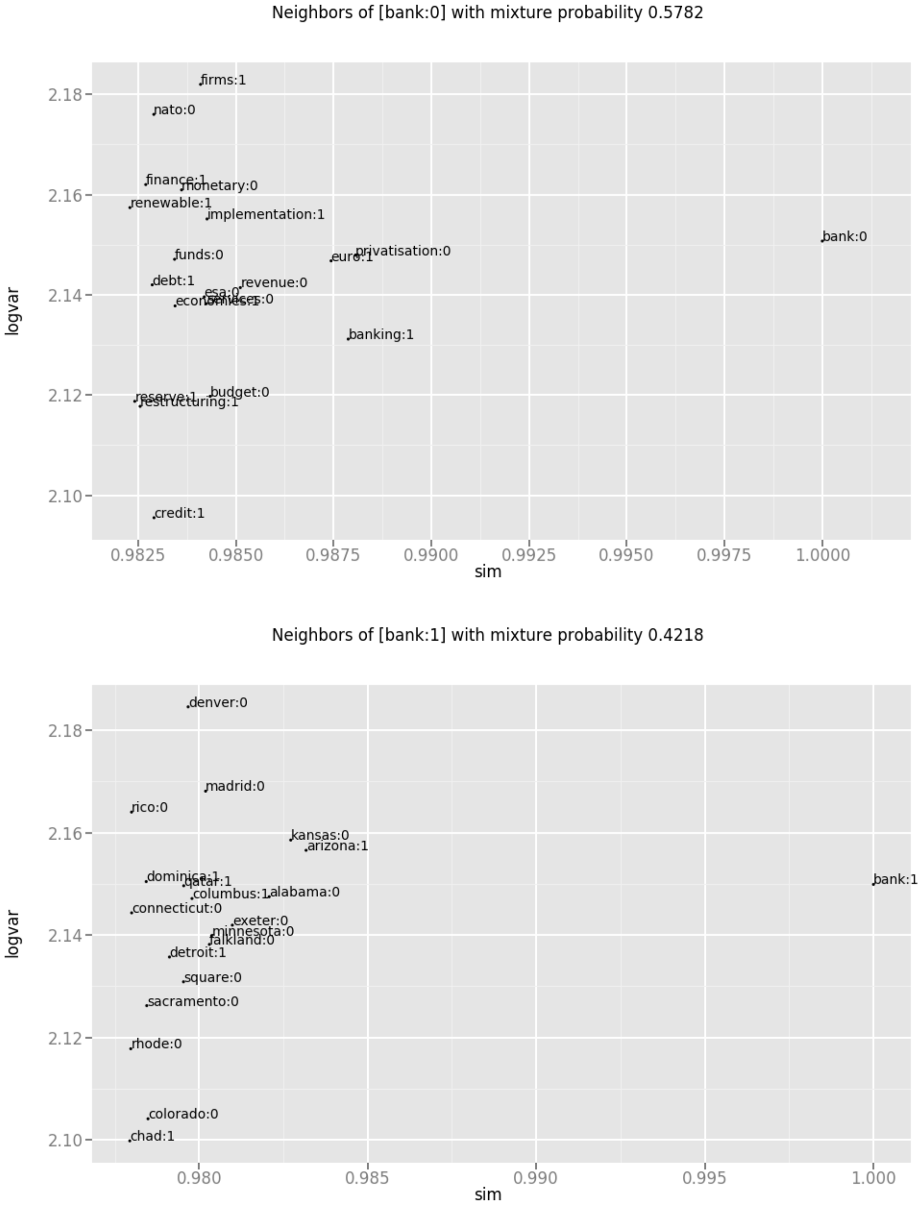
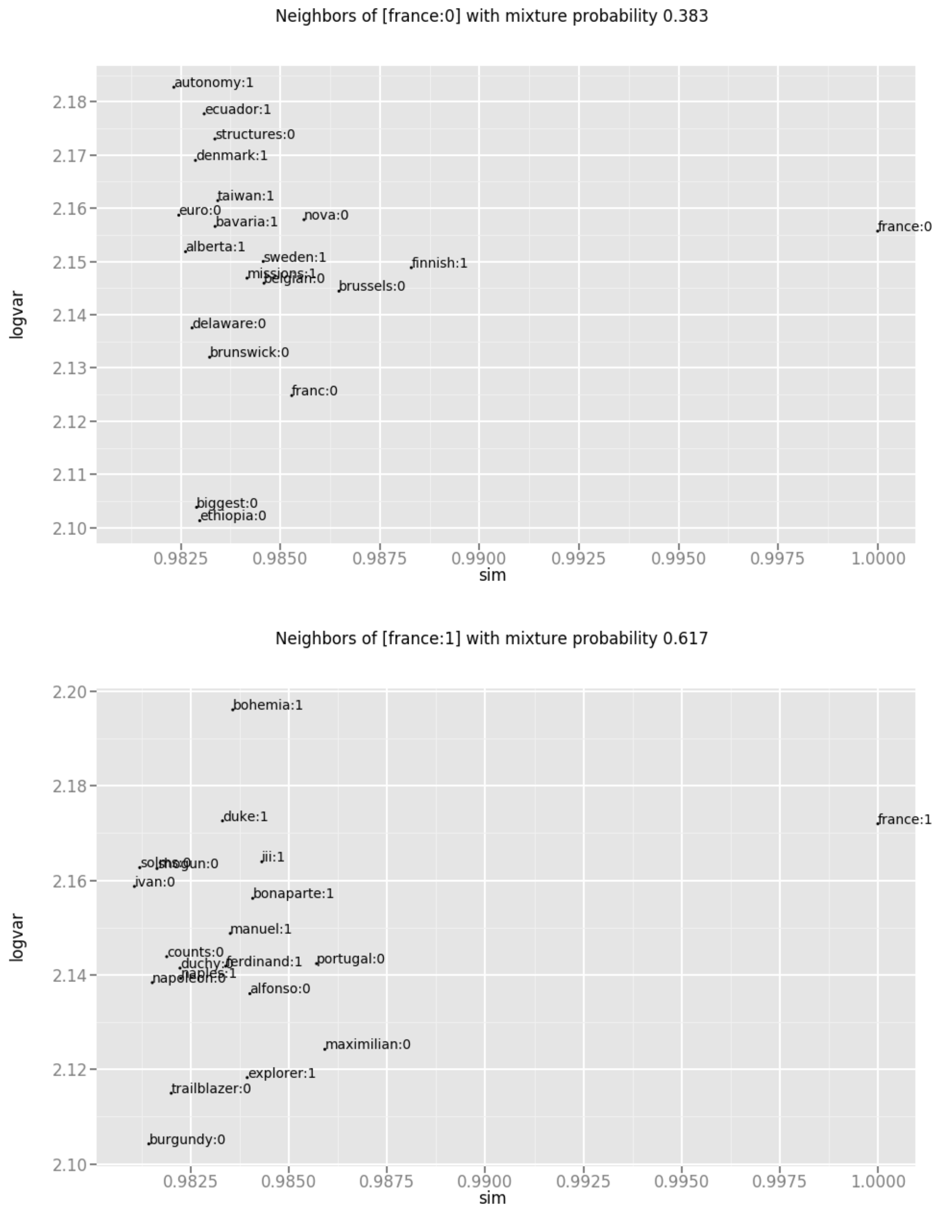


Figure 1: Highest Similarity Words for Bank (both contexts)

Word	Context	Nearest Neighbors
France	0	’france:0’, ’finnish:1’, ’brussels:0’, ’nova:0’, ’franc:0’, ’belgian:0’, ’sweden:1’, ’missions:1’, ’taiwan:1’, ’bavaria:1’
France	1	’france:0’, ’finnish:1’, ’brussels:0’, ’nova:0’, ’franc:0’, ’belgian:0’, ’sweden:1’, ’missions:1’, ’taiwan:1’, ’bavaria:1’

Table 2: Highest Similarity Words for France


Neighbors of [france:0] with mixture probability 0.383

Neighbors of [france:1] with mixture probability 0.617

Figure 2: Highest Similarity Words for France (both contexts)

Word	Context	Nearest Neighbors
Islam	0	islam:0, baptist:1, fathers:0, canonical:1, judaism:1 religions:1, eusebius:1, baptism:1, circumcision:1, anglican:1
Islam	1	islam:1, muslim:1, arabs:0, arab:0, sunni:1 arabia:1, islamic:1, arabic:0, tribal:1, syria:0

Table 3: Highest Similarity Words for Islam

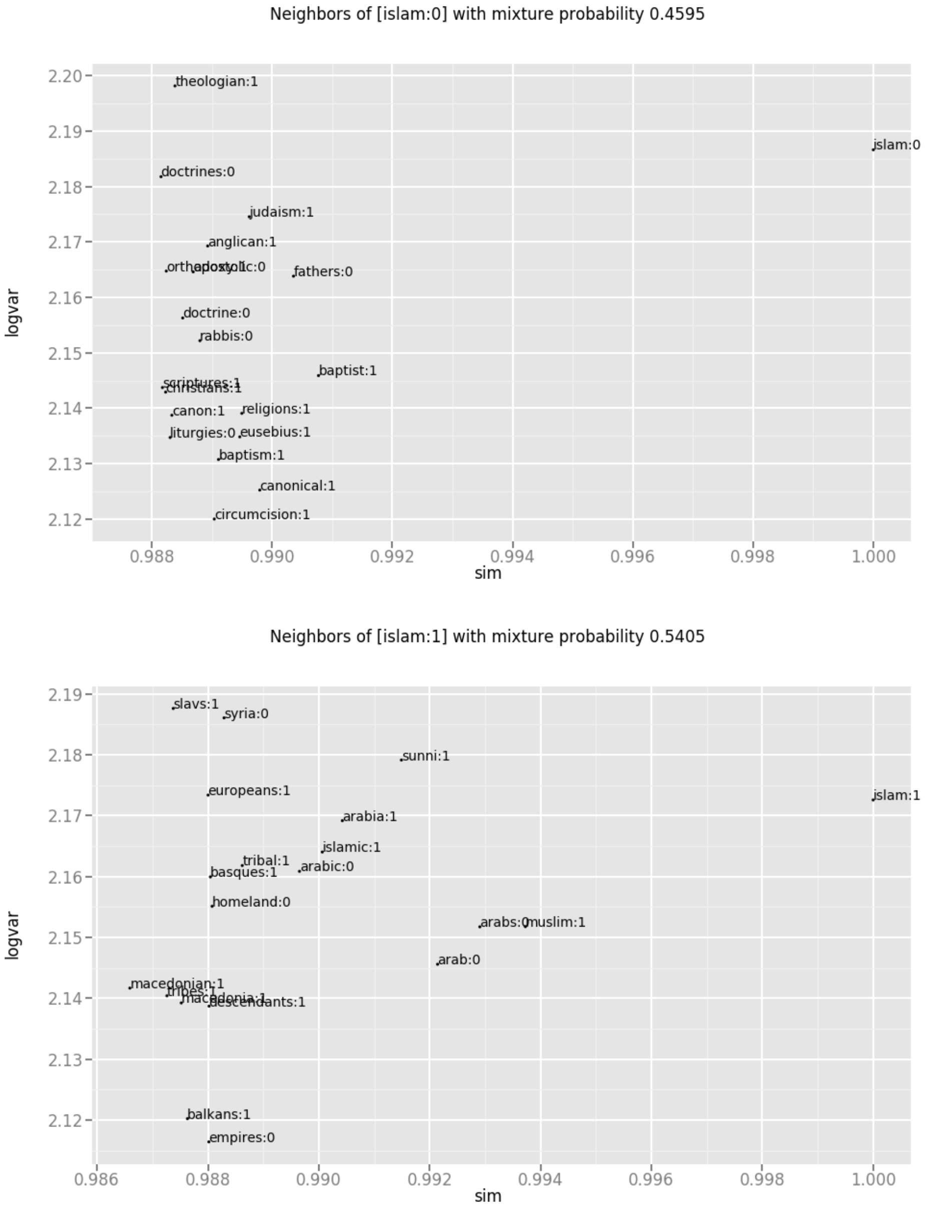


Figure 3: Highest Similarity Words for Islam (both contexts)

5 Future Work Possible

The results look good enough for second approach and working on following things may improve them further:

- The original tnesorflow implementation of the paper that we modified was not optimized. Despite our approaches having lesser number of parameters, they both are taking more time than the original approach. So a restructuring/optimization of code is required.
- Currently we are using the same dataset and vocabulary as used in original approach. It is anticipated that if we reduce the vocabulary size, the original approach will overfit because of too many parameters and our approach is supposed to work better in that case. We need to find an appropriate vocabulary and compare all the approaches
- We can work on the task of learning the hyperparameter K (number of Senses per word) instead of the current approach where we pre-assign some value to K.

6 Acknowledgments

We would like to acknowledge Prof. Piyush Rai for his constant help and pointers which paved our way to the progress we made in this project.

References

[1] Mikolov Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. ”Efficient estimation of word representations in vector space.” *arXiv preprint arXiv:1301.3781 (2013)*

[2] Luke Vilnis and Andrew McCallum. ”Word representations via gaussian embedding”. *CoRR abs/1412.6623*

[3] Ben Athiwaratkun, Andrew Gordon Wilson. ”Multimodal Word Distributions”. *arXiv preprint arXiv:1704.08424 (2017)*

[4] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. ”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, EMNLP 2014*, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pages 1059-1069.

[5] Jeffrey Pennington and Richard Socher and Christopher D. Manning. ”GloVe: Global Vectors for Word Representation”. *Empirical Methods in Natural Language Processing (EMNLP) 2014* , 1532-1543, http://www.aclweb.org/anthology/D14-1162.

[6] Princeton University ”About WordNet.” *WordNet. Princeton University. 2010.* , http://wordnet.princeton.edu