



# Paragraph Vector Based Topic Model for Language Model Adaptation

Wengong Jin, Tianxing He, Yanmin Qian, Kai Yu

Key Lab of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering  
SpeechLab, Department of Computer Science and Engineering  
Shanghai Jiao Tong University, Shanghai, China

{ascii991218, cloudygoose, yanminqian, kai.yu}@sjtu.edu.cn

## Abstract

Topic model is an important approach for language model (LM) adaptation and has attracted research interest for a long time. Latent Dirichlet Allocation (LDA), which assumes generative Dirichlet distribution with bag-of-words features for hidden topics, has been widely used as the state-of-the-art topic model. Inspired by recent development of a new paradigm of distributed paragraph representation called *paragraph vector*, a new topic model based on paragraph vector is proposed in this work. During training, each paragraph is mapped to a unique vector in continuous space. Then unsupervised clustering is performed to construct topic clusters. Topic-specific LM is then built based on clustering results. During adaptation, topic posterior is first estimated using the paragraph vector based topic model and new adapted LMs are constructed by interpolating the existing topic-specific models using topic posteriors. The proposed topic model is applied for N-gram LM adaptation and evaluated on Amazon Product Review Corpus for perplexity and a Chinese LVCSR task for CER evaluation. Results show that the proposed approach yields 11.1% relative perplexity reduction and 1.4% relative CER reduction over N-gram baseline, outperforming LDA based method proposed by previous work.

**Index Terms:** language model adaptation, representation learning, topic model

## 1. Introduction

Various approaches have been introduced to do language model (LM) adaptation for speech recognition [1] [2]. One of the idea is to incorporate topic information for each sentence or paragraph when building LM [3] [4] [5]. Topic models, represented by PLSA and LDA, are thus introduced to model the topic composition of each paragraph by inferring topic mixture (or distribution) for each paragraph. Texts of similar topic are gathered together on which topic-specific LMs are learned. The key advantage of topic model based approach for LM adaptation is that in multi-pass decoding framework, the background LM, which is built on the entire training corpus, can be automatically adjusted by interpolating topic-specific LMs according to topic composition for each hypothesis after first-pass results.

Despite the success of PLSA and LDA based LM adaptation, these topic models are limited to bag-of-words feature representation that are not capable of capturing word order information of a paragraph or even complex semantics. Recently, distributed representation [6] [7] of words and phrases (which is called **word vector**) has been introduced and proved to be

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and JiangSu NSF project No. 201302060012.

Similarity	<i>Good</i> $\approx$ <i>Well</i> ; <i>High</i> $\approx$ <i>Tall</i>
Analogy	<i>King - Man + Women</i> $\approx$ <i>Queen</i> <i>Segmention - Segment + Inform</i> $\approx$ <i>Information</i>

Table 1: Semantic relations captured by word vectors.  $a \approx b$  means word vector  $b$  is closest to  $a$  among all words.

much better than one-of-K encoding in various tasks in NLP. The basic idea is to map each word or phrase into a unique vector in the continuous space called Vector Semantic Model (VSM). The key advantage of this approach is that semantic relations such as synonym and analogy can be simply expressed by vector arithmetic. Table 1 gives some illustrations. Now a variety of algorithms have been developed for learning algorithm of word vectors such as skip-gram [8], hierarchical log by linear (HLBL) [9] and GloVe [10].

Fortunately, some of these learning algorithms have been generalized to sentences and documents, resulting in **Paragraph Vector** (PV) [11] that learns the distributed representation of sentences and paragraphs. When training paragraph vector, each word is represented by a vector in continuous space instead of one-of-K encoding and word order information is also taken into account during training. In practice, paragraph vector has outperformed LDA based feature representation in supervised classification tasks like sentiment analysis and information retrieval. Therefore, paragraph vector based topic model is more promising than previous approach for language model adaptation. In-depth research on this issue will be provided.

The original contributions of this paper are:

- A new LM adaptation framework based on paragraph vector based topic model is developed.
- Comparison of LM adaptation based on LDA and paragraph vector is given, justifying that paragraph vector based approach outperforms LDA in terms of both perplexity and character error rate for speech recognition.

The whole paper is organized as follows: Section 2 introduces topic model based LM adaptation framework. Section 3 describes Paragraph Vector, including the feature learning algorithm and topic mixture estimation method. Section 4 evaluates and compares perplexity and speech recognition results for different adaptation methods.

## 2. Topic model based LM adaptation

In the framework of topic model based LM adaptation, topic mixture  $\theta^{(s)}$  is inferred for each paragraph  $s$  by topic model  $\mathcal{T}$ , which is learned from training corpus in an unsupervised fashion. The training corpus  $\mathcal{C}$  is then partitioned into  $K$  subsets  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$  where  $\mathcal{C}_i$  is the set of paragraphs with mixture

weight of topic  $i$  higher than other topics. Afterwards, background LM  $\mathcal{M}$  is trained on the whole training set  $\mathcal{C}$  and topic-specific LMs  $\mathcal{M}_i$  are learned on  $\mathcal{C}_i$ . During testing, there are two schemes of interpolating background LM and topic-specific LMs which have been investigated in previous work [4].

### 2.1. Hard interpolation

In this setting, each paragraph  $s$  in test set is labelled as topic  $k$  where  $k = \arg \max_j \theta_j^{(s)}$ . The language model probability of the  $i^{\text{th}}$  word in paragraph  $s = x_1 x_2 \cdots x_l = x_1^l$  is:

$$P(x_i | x_1^{i-1}) = C_B^{(k)} P_{\mathcal{M}}(x_i | x_1^{i-1}) + (1 - C_B^{(k)}) P_{\mathcal{M}_k}(x_i | x_1^{i-1}) \quad (1)$$

where  $C_B^{(k)}$  is topic-specific background LM weight. This interpolation strategy focus on using topic-specific weights which can be chosen by cross-validation.

### 2.2. Soft interpolation

A problem of hard interpolation scheme is that the probability of inferior topics  $j \neq k$  for each paragraph is ignored, which may cause degradation because a paragraph may have multiple topics involved. Therefore, instead of assigning each paragraph to a topic with hard decision, we interpolate all topic-specific LMs according to mixture weights in  $\theta$ . Now the probability of the  $i^{\text{th}}$  word  $P(x_i | x_1^{i-1})$  becomes

$$C_B P_{\mathcal{M}}(x_i | x_1^{i-1}) + (1 - C_B) \sum_{i=1}^K \frac{\theta_k}{\sum_{i=1}^K \theta_i} P_{\mathcal{M}_k}(x_i | x_1^{i-1}) \quad (2)$$

in which the background LM weight  $C_B$  is shared among all topics and chosen by cross-validation.

## 3. Paragraph vector

This section describes learning algorithm of paragraph vector, which is generalized from skip-gram [8]. In general, paragraph vector is a concatenation of two vectors from two representation learning methods: *Distributed Memory* (PV-DM) and *Distributed Bag of Words* (PV-DBOW). Mathematically, the  $i^{\text{th}}$  paragraph is mapped to a unique vector  $\mathbf{d}_i = [\mathbf{d}_i^{(1)}, \mathbf{d}_i^{(2)}]$ , where  $\mathbf{d}_i^{(1)}$  is the PV-DM vector, a column in PV-DM matrix  $\mathbf{D}^{(1)}$  and  $\mathbf{d}_i^{(2)}$  the PV-DBOW vector, a column in PV-DBOW matrix  $\mathbf{D}^{(2)}$ . These matrices are learned separately by two different algorithms.

### 3.1. Distributed Memory

Suppose each paragraph  $s$  is represented as a word sequence  $x_1 x_2 \cdots x_l$ , from which a set of sliding windows are sampled, denoted as  $x_{i_1} x_{i_2} \cdots x_{i_t}$  of fixed length  $t$ . PV-DM aims at maximizing the probability  $P(x_{i_{t+1}} | x_{i_1} x_{i_2} \cdots x_{i_t})$  for all such sliding windows where the probability is modelled by following neural network function illustrated by Figure 1

$$P(x_{i_{t+1}} | x_{i_1} x_{i_2} \cdots x_{i_t}) = \frac{\exp(y_{i_{t+1}})}{\sum_k \exp(y_k)} \quad (3)$$

$$\mathbf{y} = \mathbf{b} + \mathbf{U}^{(1)} \left( \mathbf{d}_s^{(1)} + \sum_{k=1}^t \mathbf{W}(x_{i_k}) \right) \quad (4)$$

where matrix  $\mathbf{U}^{(1)}$  is the output matrix and  $\mathbf{W}(x_{i_k})$  is  $x_{i_k}$ 's word vector. All these matrices including  $\mathbf{D}^{(1)}$  are randomly initialized and then the objective function is optimized using stochastic gradient descent (SGD). Note that from above equations, paragraph vector  $\mathbf{d}_s^{(1)}$  is shared over all sliding windows in paragraph  $s$ , but not in other paragraphs. Therefore,  $\mathbf{d}_s^{(1)}$  will encode semantic information for paragraph  $s$  exclusively.

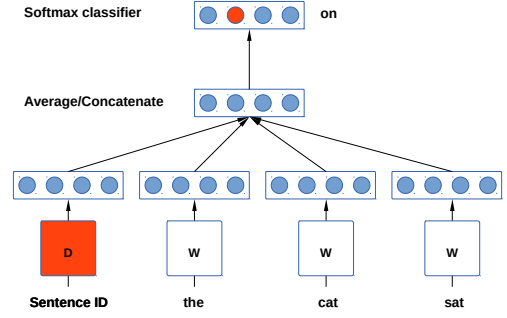


Figure 1: Paragraph Vector: Distributed Memory

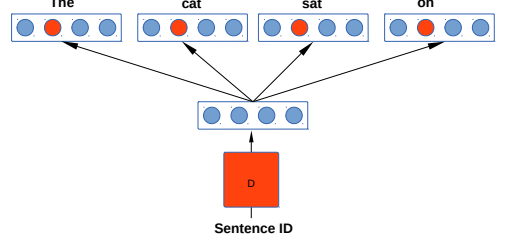


Figure 2: Paragraph Vector: Distributed Bag of Words

In practice, the softmax layer is substituted by hierarchical softmax for ease of computation. The size of sliding window and the dimension of  $\mathbf{D}^{(1)}$  and  $\mathbf{W}$  is chosen by cross-validation.

### 3.2. Distributed bag of words

Similar to PV-DM framework, sliding window of words  $x_{i_1} x_{i_2} \cdots x_{i_t}$  is sampled over the paragraph  $s$ . However, the objective function changes to maximize the average log probability  $\frac{1}{t} \sum_{k=1}^t \log P(x_{i_k} | \mathbf{d}_s^{(2)})$  given only paragraph vector  $\mathbf{d}_s^{(2)}$ , in which probability is calculated by softmax function with weight matrix  $\mathbf{U}^{(2)}$ . Again the objective function is optimized by SGD with matrix  $\mathbf{D}^{(2)}$  and  $\mathbf{U}^{(2)}$  randomly initialized.

### 3.3. Inference stage

After training stage, matrices  $\mathbf{D}^{(1)}$ ,  $\mathbf{D}^{(2)}$  are learned whose column serves as the PV-DM and PV-DBOW vector of paragraphs in the training set. However, during the inference stage, representation for new paragraphs remains unknown.

The solution is again to apply above two learning methods with all learned parameters  $\mathbf{U}^{(1)}$ ,  $\mathbf{U}^{(2)}$  and  $\mathbf{W}$  fixed. Only vectors for these new paragraphs are learned by SGD. After optimization converges, these vectors become the representation of new paragraphs.

### 3.4. Paragraph vector based LM adaptation

To build topic-specific LMs, K-means algorithm is applied to partition the training set into different clusters using learned paragraph vectors. The similarity measure of two paragraphs is the **cosine distance** of their paragraph vectors, which consistently outperforms euclidean distance in our experiment. After clustering stage, centroid feature vectors of  $K$  clusters  $\xi_1, \xi_2, \cdots, \xi_K$  are acquired. The topic mixture  $\theta^{(s)}$  of paragraph  $s$  is defined as  $\theta_k^{(s)} = \frac{f(\cos(\xi_s, \xi_k))}{\sum_{i=1}^K f(\cos(\xi_s, \xi_i))}$  where  $f$  is

a scale function. In practice, we find that  $f(x) = \max(0, x)^2$  performs better than our heuristics.

### 3.5. Advantage of paragraph vector

Paragraph vector addresses some key weaknesses of BOW features. First, paragraph vectors encode the word order in the same way as N-grams. This is important because word order preserves much more information of a paragraph than BOW features. Another advantage is that paragraph vectors

are trained with word vectors, which allows them to inherit nice properties of word vectors. More importantly, the learning algorithm of paragraph vector can be easily extended by techniques like multi-task learning. Finally, by mapping paragraphs to continuous space, it would be easier than LDA to apply classical machine learning algorithms like clustering or classification.

The key difference between LDA and paragraph vector based topic model is that topic mixture weights given by paragraph vector are based on discriminative criterion instead of generative likelihood given by LDA. In fact, during soft interpolation, interpolating all topic-specific LMs does harm the performance of LM adaptation when using paragraph vector while LDA does not. Therefore, only 2 topics-specific LMs with highest score is chosen when doing soft interpolation.

## 4. Experiments

Various topic model based LM adaptation schemes are evaluated in this section. First, perplexity results are reported on *Amazon Product Review Corpus (APRC)*, a set of product reviews on various items. Each review may contain one or more paragraphs with average length of 200 words. Speech recognition results are given by another experiment on a Chinese LVCSR task where each utterance is of length 10 on average.

### 4.1. APRC Experiments

#### 4.1.1. Perplexity results

Our language model adaptation scheme is first evaluated on the subset of APRC. The corpus is divided into five categories: books, music, DVD, electronics and kitchen housewares. The corpus for each topic is divided into training, validation and test sets with ratio 8:1:1. The vocabulary size is about 71k with OOV rate 0.74% on the test set. Back-off 4-gram model with

	Topic	Hard	Soft	Perplexity
Background LM	—	—	—	135.8
Oracle Adapted LM	5	✓	—	124.1
LDA Adapted LM	50	✓	—	125.7
		—	✓	123.9
PV Adapted LM	50	✓	—	122.2
		—	✓	<b>120.8</b>

Table 2: Results of various LM on APRC

Kneser-Ney smoothing (*Background LM*) is built on APRC and the perplexity results (OOV not included) is reported. In addition, hard interpolation scheme can be performed by incorporating prior knowledge about topic partitions, yielding *Oracle Adapted LM*. To perform LM adaptation, each review is considered as a single paragraph for both LDA and PV. If one review has multiple paragraphs, they are concatenated one-by-one separated by special symbol. Paragraph vectors are of dimension 600. As in Table 2, both model performs better than baseline while paragraph vector based adaptation outperforms LDA. The best result yields 11.1% relative improvement over background LM and additional 2.5% over oracle adapted LM and LDA.

#### 4.1.2. Discussion: hyper-parameters

All hyper-parameters like number of topics  $K$ , background LM weights  $C_B^{(i)}$  (hard) and  $C_B$  (soft) are tuned on validation set. In this section, the impact of tuning hyper-parameters on the performance of adapted LM is investigated. As in Figure 3, increasing number of topics consistently yields better performance and soft interpolation scheme outperforms the other in most cases, which empirically proves that dynamic interpolation for each paragraph is more efficient.

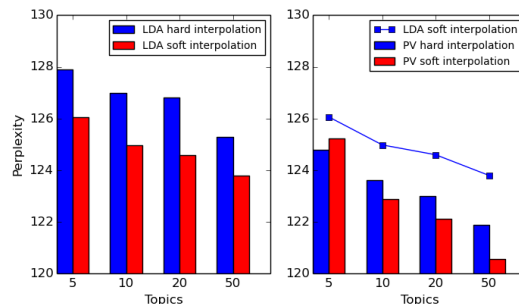


Figure 3: Cross validation results with different interpolation schemes as well as number of topics.

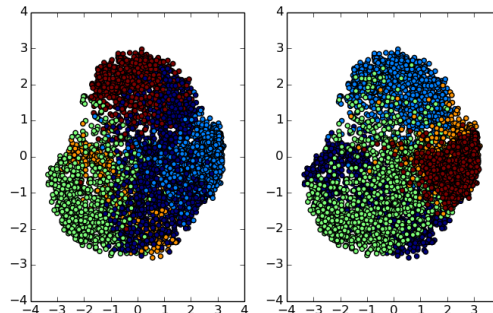


Figure 4: Data visualization via t-SNE. Note that labels in both pictures are not aligned.

#### 4.1.3. Data visualization

Figure 4 gives a visualization of a subset of APRC test set. All points in both picture represents a paragraph in the test set, whose location is determined by projecting its paragraph vector to the plane using t-SNE [18]. Different colors indicate different topic labels for each paragraph. In the picture on the left, the label is predicted by K-means clustering (number of topics is 5) while they are given by ground truth in the other picture. In comparison, topic mixtures inferred from paragraph vector is quite similar to the ground truth. In addition, adding number of topics could disentangle some overlapping classes, which shed lights on the result in Figure 3.

#### 4.1.4. Topic classification accuracy

In order to give a in-depth comparison of LDA and paragraph vector, a supervised classification task is constructed in which each review  $s$  is represented by either LDA topic mixture  $\theta_s$  or paragraph vector. The label for each review is given by prior topic partition, which is in total 5 labels as mentioned at the beginning. Linear SVM is then trained using either 1% or all of training data and classification accuracy on test set is reported. Again, paragraph vector significantly outperforms LDA, justifying the discriminative nature of paragraph vector.

	Amount of training data	Classification accuracy
LDA	1%	76.46%
	100%	76.55%
PV	1%	<b>90.09%</b>
	100%	<b>92.48%</b>

Table 3: Feature comparison in terms of topic classification

## 4.2. Speech Recognition Experiment

In this section, adapted LMs are evaluated over a Chinese LVCSR task. The training data is a corpus of smart phone messages, with 8 million sentences and 48 million tokens in total. The test set consists of 3-hour transcribed smart-phone audio data (3k utterances).

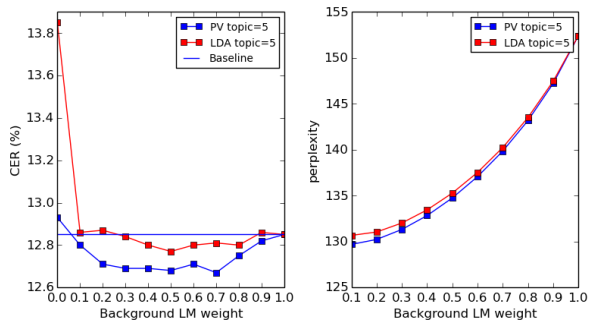


Figure 5: CER and perplexity results on test set. Left: CER results under different  $C_B$ . Right: Perplexity results under different  $C_B$

The acoustic model is a context-dependent DNN-HMM [12] with 6021 tied triphone states, which is trained on 600 hours of audio data collected from smart-phones. 13-dimensional PLP with its first and second derivatives is used as acoustic feature. In total 11 frames, 5 on the right and 5 on the left, are used as the input to DNN. An RBM initialized DNN with six 2048-node hidden layers is then trained using stochastic gradient descent. Back-off N-gram model with Kneser-Ney smoothing is used as language model for generating lattice. A LM scaling factor of 25 is used for all experiments and 30-best lists are used for re-scoring with different language models. In

	Topics	Optimal $C_B$	Perplexity	CER(%)
Baseline	—	—	152.42	12.85
Oracle	—	—	—	4.43
PV	5	0.7	139.85	<b>12.67</b>
	10	0.7	133.85	12.79
	15	0.7	129.66	<b>12.67</b>
LDA	5	0.5	135.28	12.77
	10	0.9	143.99	12.77
	15	0.9	140.70	12.75

Table 4: N-best rescoring results. Best CER over test set is given under optimal  $C_B$ . Perplexity is calculated under the same setting with CER.

table 4, 4-gram re-scoring baseline results are shown. For both LDA and paragraph vector based LM adaptation, only results under soft interpolation scheme on test data is reported due to space limitation. Different from APRC, each utterance in both training and test set contains only one sentence. Thus each sentence is regarded as a single paragraph. As a result, both paragraph vector and LDA suffer from slight improvement over baseline. The best result is achieved by paragraph vector when the number of topic is 5, which yields 1.4% relative improvement over baseline.

#### 4.2.1. Discussion and analysis

Figure 5 shows obvious inconsistency between perplexity and CER dynamics under different background LM weights, as reported in previous work [4] where the author attribute this problem to erroneous N-best hypotheses distorting the topic inference. A justification is given by Table 5 where topic mixture is inferred from clean reference during re-scoring. As in Table 5, both methods improves dramatically under this case.

Although LDA outperforms paragraph vector when topic mixture is inferred from clean text, paragraph vector still has advantages over LDA because paragraph vector is trained based on discriminative criterion and thus its topic inference is more robust under small variations over the text. To support this con-

Topics	PV		LDA	
	Perplexity	CER	Perplexity	CER
5	129.85	12.44	130.67	12.28
10	122.26	12.28	117.63	11.88
15	117.38	12.10	109.07	11.66

Table 5: N-best rescoring results: topic inference from clean reference. The optimal  $C_B$  is 0.1 for both methods.

jecture and further investigate what kind of sentences can get most improvement from our adaptation approach, a histogram of CER improvement over baseline CER for each utterance is drawn. The x-axi is uniformly divided into 20 parts, each denoting a category with baseline CER in a particular range. Utterances of higher baseline CER have more erroneous hypotheses. As in Figure 6, erroneous hypotheses are improved by adapted LM while correct hypotheses (CER < 5%) got corrupted in both methods. Furthermore, LDA suffers much more from this problem than paragraph vector, supporting the robustness of paragraph vector based adaptation under small variations.

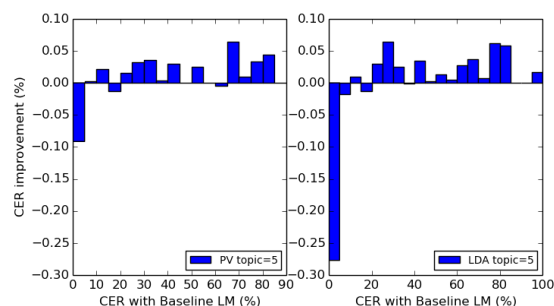


Figure 6: CER improvement statistics. X-axis: The CER given by baseline LM. Y-axis: CER improvement over baseline.  $C_B = 0.5$  for LDA and 0.7 for paragraph vector.

#### 4.2.2. Limitation of Paragraph Vector

As in Table 5, LDA outperforms paragraph vector also in perplexity measure, a contradiction to previous results on APRC. We attribute this problem to the fact that in this dataset, each utterance is so short that each paragraph vector cannot get enough updates during training (average length  $\approx 10$  compared to 200 in APRC). To support this claim, a supplementary experiment is conducted on APRC where each review is split into sentences and each sentence is considered as a paragraph instead of the whole review. The perplexity results become 116 for LDA and 120 for paragraph vector when number of topic is 10. Therefore, although paragraph vector fails to outperforms LDA when paragraphs are short, it is still suitable to long context-span language model adaptation.

## 5. Conclusion and future work

In this paper, unsupervised LM adaptation methods based on LDA and paragraph vector are investigated and compared. Both methods yields better perplexity results over baseline in APRC while paragraph vector outperforms LDA. For speech recognition experiments, paragraph vector gives better result than LDA in N-best re-scoring because of its robustness of topic inference. For future work, we will focus on improving paragraph vector's performance on short paragraphs or even single sentence case so that paragraph vector based adaptation methods is efficient under different situations. Another direction of our work is to improve the robustness of topic inference for topic-model based methods, especially for LDA.

## 6. References

- [1] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech & Language*, vol. 10, no. 3, pp. 187–228, 1996.
- [2] R. M. Iyer and M. Ostendorf, "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 30–39, 1999.
- [3] Y. Tam and T. Schultz, "Dynamic language model adaptation using variational bayes inference." in *INTERSPEECH*, 2005, pp. 5–8.
- [4] A. Heidel, H. Chang, and L. Lee, "Language model adaptation using latent dirichlet allocation and an efficient topic inference algorithm." in *INTERSPEECH*, 2007, pp. 2361–2364.
- [5] Y. Liu and F. Liu, "Unsupervised language model adaptation via topic modeling based on named entity hypotheses," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4921–4924.
- [6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, 1988.
- [7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
- [9] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2265–2273.
- [10] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, 2014.
- [11] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*, 2014.
- [12] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.
- [13] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.
- [14] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
- [15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.
- [16] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proceedings of the international workshop on artificial intelligence and statistics*. Citeseer, 2005, pp. 246–252.
- [17] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, 2012.
- [18] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.
- [19] A. Stolcke *et al.*, "Srlm-an extensible language modeling toolkit." in *INTERSPEECH*, 2002.
- [20] G. Karypis, "Cluto-a clustering toolkit," DTIC Document, Tech. Rep., 2002.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [22] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, <http://is.muni.cz/publication/884893/en>.