Text Representation for Machine Learning Applications

Aastha Nigam*
Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556
anigam@nd.edu

Pingjie Tang*
Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556
ptang@nd.edu

Henry K. Dambanemuya*
Kroc Institue for International Peace Studies
University of Notre Dame
Notre Dame, IN 46556
hdambane@nd.edu

Bryan (Ning) Xia*
Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556
nxia@nd.edu

ABSTRACT

Most of the data generated today is unstructured which typically consists of text information. Understanding such text data is imperative given that it is rich in information and can be used widely across various applications. However, the key to understanding such data is its representation. In this survey, we discuss various text representation methods starting from count based methods to state of the art methods including distributed representational learning. These algorithms can transform large volumes of text into effective vector representations capturing the same semantic information. Further, such representations can be utilized by various machine learning algorithms.

1 INTRODUCTION

Data is growing at an exponential rate, where unstructured data is growing significantly faster than structured data. Unstructured data is typically text heavy. Text based data is prevalent in varied domains whether it be social media with users sharing their opinions [64] or articles published by media companies or clinical notes for patients in the hospital and online reviews given by users expressing their preferences to some businesses.

Text data being rich in information gives us a unique opportunity to derive valuable insights which might not be comprehensible from quantitative data [15, 129]. Consequently, the main objective of various natural language processing (NLP) algorithms is to obtain human-like understanding of text [80]. For example, many researchers aim to mine the opinion of users about a restaurant from their online reviews or predict public sentiment from social media about a political event. Over the years text has been used in various applications such as email filtering [25, 31, 137], document organization [13, 54, 65, 81, 120, 121, 142], sentiment prediction [103], opinion mining [103, 133], polarization detection [29, 123], topic inference [19, 90], text summarizing [12, 49, 88], anomaly detection [15], language translation, question answering [53], content mining [2] and many more.

However, being unstructured content it adds complexity to model, decipher automatically or use in conjunction with traditional features for a machine learning framework [57]. Moreover, even though large of volumes of text information is widely available and can be leveraged for interesting applications, it is rife with problems [2]. Like most data, it suffers from traditional problems such as class imbalance and lack of class labels, but in addition there are some inherent issues with text information. Apart from being unstructured, text mining and representation learning becomes more challenging due to the following discussed factors [129]:

- **Noise**: Text can be very noisy with various spelling mistakes, colloquialisms, slang words and emoticons. This affects the quality and reliability of the data.
- **Ambiguity**: Language in general can be ambiguous with same word or sentence having multiple meanings. Additionally, sarcasm is commonly observed in human generated data on social media. Even though it is easy for humans to infer the meaning, it is still a challenging task to be done automatically.
- **Semantic Structure**: The order and placement of words can drastically change the meaning. For example, the presence of 'not' can negate the meaning of a sentence.
- **Domain Knowledge**: Domain knowledge is critical for some text. In order to interpret the meaning of the sentence is it crucial to have an understanding of the domain and be familiar with the domain jargon.
- **Multilingual**: Usage of multiple languages in the same piece of information. Additionally most models are designed for English language and therefore are not generalizable to other languages.

A lot of research has been dedicated to address each of these concerns individually [38, 44, 126]. However, in this survey we focus on how text can be represented as numeric/continuous vectors for easier representation, understanding and applicability to traditional machine learning frameworks. Text can be seen as a collection of entities such as documents, sentences, words or characters and most algorithms leverage the implicit relationship between these entities to infer them as vectors.

Over the years, many methods and algorithms have been used to infer vectors from text be at character, word, sentence or document level. All the methods are aimed at better quantifying the richness in the information and making them more suitable for machine learning frameworks such as to perform clustering, dimensionality reduction or text classification. In this survey, we study how text representation methods have evolved from manually selecting the

*Equal Contribution

features called feature engineering to more state of the art representational learning methods which leverage neural networks to discover relevant embeddings.

Contribution and Organization In this paper, we present a comprehensive study of various text representation methods starting from bag of words approach to more state of the art representational learning methods. We describe various commonly used text representation methods and their variations and discuss various text mining applications they have been used in. We conclude with a discussion about the future of text representation based on our findings. We would like to note that this paper, strictly focuses on representation of text for machine learning frameworks and therefore uses content, data and text interchangeably.

2 FEATURE ENGINEERING

In this section, we discuss various popularly used feature engineering models such as bag of words, semantic representation and latent semantic analysis. Most methods rely on count based methods and involve manual effort to derive meaninful representations.

2.1 Bag of Words Model

The Bag of Words (BOW) or unigram model is one of the most popular methods for representing text. The model treats words as independent features. Each sentence in a BOW model is therefore represented as a multiset of its words. Using the BOW model, text documents can therefore be represented through a high-dimensional sparse vector whereby the term frequency of each feature represents each dimension. Although the multiplicity of words (word-frequency) is retained, word order, context, and grammar are lost. One common application of the BOW is to generate document term frequency features to train a classifier for document classification in information retrieval systems.

A natural extension of the BOW model is Term Frequency - Inverse Document Frequency (TF-IDF) where the term frequencies of each feature vector are discounted by the inverse document frequencies to down-weight terms common terms and identify terms that are discriminative for documents in the corpus. To preserve word order and context, the BOW model can be extended to an n-gram language model where n represents a set of consecutive words extracted from a sentence as a feature vector. In this case, the BOW language model is conceptually a special case of the n-gram language model where n = 1, hence a unigram model.

Traditional BOW representations have several limitations. For example, BOW vector representations are often high-dimensional and very sparse. Although increasing the order of n-grams can help in dimensionality reduction and improve prediction accuracy, it further exacerbates data sparsity. The BOW model also ignores word semantics and fails to capture synonymy, polysemy, and word context. For example a sentiment classifier would have to be exposed to a very large set of labelled data to learn that similar words are predictive towards similar sentiment. However, extensions of the BOW model using Latent Semantic Index (LSI), discussed later in this paper, attempt to overcome problems of word order and context by applying Singular Value Decomposition (SVD) to the BOW / TF-IDF features to find a latent semantic space that captures word synonym in the corpus. To overcome sparsity, Chen et al [27] proposed a Dense Cohort of Terms (dCOT) unsupervised algorithm that maps high-dimensional sparse BOW into low-dimensional dense representations providing a closed-form transformation of the original sparse BOW features that is extremely fast to train and apply. Their model shows that dCOT features significantly improve classification accuracy in document classification tasks. Below, we discuss some applications of the n-gram model.

Several n-gram techniques have been used for spelling error correction and detection [6, 32, 34, 55, 97, 102, 113, 135]. Zamora et al [146] used a trigram analysis for spelling error detection. The goal of their study was to determine the utility of trigram analysis in automatically detecting and correcting mispellings. Using a dataset of 50, 000 misspelt words from six different datasets, the authors' trigram analysis technique was able to accurately identify the error site within a misspelling. However, their technique could not distinguish between different error types (e.g. material, positional, and ordinal similarity) or between valid words and misspellings. While limiting their study to material similarity, the extent to which pairs of strings contain identical characters, Angell et al [8] used a tri-gram similarity measure to automate spelling correction. Using a dictionary of 64, 636 words and a collection of 1, 544 misspelt words, the authors developed a nearest neighbour search model to replace a misspelt word by a word in the dictionary which best matches the misspelling. The degree of match was calculated using a similarity coefficient based on the number of trigrams common to the two words. Their model correctly identified over 75% of the mispellings if the correct form of the word was contained in the dictionary.

Brants et al [22] used an n-gram language model trained on unlabelled monolingual text, for machine translation. In their proposed distributed infrastructure, the authors train an n-gram language model using 2 trillion tokens or terms resulting in a language model comprising 300 billion n-grams used for machine translation. Their n-gram model achieved a competitive Bilingual Evaluation Understudy (BLEU) score of 0.4535 on the Arabic - English NIST subset in the 2006 NIST machine translations evaluation. The BLEU score measures the quality of text which has been machine-translated from one natural language to another on a 0 - 1 scale. Although their evaluation system used a mixture of 5, 6, and 7 n-grams, the infrastructure is capable of scaling to larger amounts of training data and higher n-gram orders. Other n-gram applications include text searching [98], text retrieval [79], text filtering [26], dictionary lookup pattern searching in large dictionaries [104].

2.2 Semantic Representation

Most researches use bag of words model, however Scott et al.[119] presented their research to examine how text can be represented using syntactic and semantic relationships between words from text. Various representations were evaluated based on the performance of a rule based learner – RIPPER [119]. The alternative representations were proposed to alleviate problems associated with bag of words models as they broke word and sentence order and syntactic structure, semantic relationships can be missing and context can be agnostic. They discussed phrase based representations which utilized nouns, since words alone were not always represent atomic meanings. On the other hand, phrases, especially noun phrases,

carried meaning full semantic information. For example, the word "Machine" and "Learning" each can represent distinct meaning, but the phrase "Machine Learning" expresses a very specific meaning related to Artificial Intelligence. Due to the redundancy, high dimensionality and non-uniformly distribution and noise of phrase based representation, Scott et al. [118, 119] proposed two algorithms for the noun phrase extraction, namely Noun Phrase Extractor (NoPE), which employed a part-of-speech tag assignment algorithm and a noun phrase grouping algorithm. [119] examined some alternative ways to represent text based on syntactic and semantic relationships between words, WordNet[95] was used to extract the hypernym and synonym information due to its hierarchical property.

Another notable method to represent text is to group words semantically related to each other. Brown[23] clustering is a form of hierarchical clustering of words based on the semantic relatedness given a similar context. The words sequence of a input text are initially clustered as individual clusters, then these clusters are merged according to a quality maximum-likelihood estimate process. The quality is defined as the logarithm of joint probability of the input words in the context of class-based bigram language model[82]. For each word, the probability is defined as the multiplication of the probability of the word given its cluster and the probability of the word's cluster given the previous word's cluster[82]. Even though Brown clustering have been very sucessful in several NLP applications, it fails to consider the word usage in a wider context due to the nature of relying only on the bigram statistics[134].

WordNet[94, 95] is an on-line lexical reference system, which organizes English nouns, verbs and adjectives into synonym sets. WordNet is used as center resource for various semantic relatedness representation methods[24]. The co-occurance information of the raw text is used to measure the semantic relatedness by combining the structure and content of WordNet[106].

2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) is a technique in natural language processing, which is primarily utilized to describe the semantic content in contextual data through exploring the structure in word usage across documents. To this end, a sparse and high-dimensional term-document matrix M is built whose rows and columns correspond to words and documents respectively, each entry represents the occurrences of the term in current document. Singular Vector Decomposition (SVD) technique is applied on such matrix [42, 75] in order to get a low-rank (say rank = k) approximation to matrix. Thus, each row and column is mapped in the form of vectors to a k-rank LSI space that is defined by k largest eigenvectors of MM^T [89]. Cosine similarities can be used to compute similarities between words vectors. Another advantage of mapping term vector on lower dimensional space is it partially alleviates the inability of vector space model to identify synonymy and polysemy. Because some components of polysemy words can be preserved and the dimensions associated with similar meaning terms are merged as well.

LSA has been integrated into and provides elegant solutions for numerous applications across text classification, data mining and information retrieval. For instance, [145] utilized LSA to reduce noise during the training stage. Software engineering recasted concept location problem by treating source code as text and software elements as terms and used LSA to index software elements to provide efficient search [108]. [147] performed SVD on an expanded term-document matrix that includes both training data and background text to classify text. [36] proposed to used LSA to represent queries and documents in a latent semantic space to solve high dimension data problem in information retrieval.

Even LSA achieves great success across a broad range of tasks, however, potential limitations still exist. First of all, LSA exhibits its inability in capturing propositional meaning as it is blind to word order. Also LSA representations are unable to understand higher-order word combinations such as phrases, clauses and sentences. Secondly, LSA cannot completely capture synonumy and polysemy. Because a certain word will be thought as carrying the identical meaning if it occurs same number of times in a document.

3 FEATURE ENGINEERING V/S LEARNING

Extracting feature representation is a crucial step in most machine learning tasks. Choosing discriminating and informative features usually would improve learning and increase interpretability. Traditional feature engineering methods usually heavily rely on domain knowledge and show inability of extracting discriminative data features. Even though domain knowledge can compensate the weakness to some extent, however, it still requires extremely expensive human labor. Furthermore, it will weaken the applicability of machine learning algorithms as the more domain knowledge involved in the phase of feature learning, the more learning methods will be confined within a specific task. On the other hand, real world raw data is usually complex, noisy and has various formats such as images, videos, audios and texts, which poses tremendous challenges to the conventional hand-crafted and domain-dependent feature engineering methods. These challenges motivate people to come up ways to design techniques that efficiently learn features automatically, less dependent on people, and guarantee feature quality at the same time.

Representation learning, also known as feature learning, refers to a set of techniques that use machine to learn feature representations that suit for and computationally convenient to machine learning tasks. Representational learning is obviously different from hand-crafted feature engineering and can be categorized into supervised, semi-supervised and unsupervised learning. Supervised representation learning learns feature representations by labeled data, representative methods such as supervised dictionary learning [87], which learns a dictionary of representative elements, that is, features, by exploiting both input data structures and labels, such that each data point can be represented as a weighted sum of the features. Supervised multilayer neural networks is another common method to perform feature learning. They use hidden layers to learn feature representations for input which are subsequently used for learning tasks at the output layer. Not large volume of references could be found that focusing on using supervised learning algorithms to perform representation learning tasks, the reasons can be explained as following, in the real world, the amount of labeled data are relatively limited for people to access, acquiring

data labels can be very labor-intensive and time-consuming. Therefore the demands of exploring representation learning approaches that can benefit from unlabeled data has been gradually attracting more attention. Semi-supervised learning combines large set of unlabeled data with, usually very smaller set of, labeled data to gain better feature representations [11]. Its wide applications can be found in the domain of text classification. [101] explored the use of generative models for semi-supervised learning in the text classification area. [4] proposed semi-supervised subspace clustering algorithm to address a text classification problem. [141]proposed a semi-supervised representation learning approach in a cross-lingual text classification problem. Semi-supervised feature learning also been widely utlized in other research areas. A. Grover el al. applied semi-supervised learning for scalable feature learning on networks [52]. [112] applied semi-supervised method to extract more compact representations on top of bag of words representations in a text recognition problem. [45] proposed a semi-supervised multi-feature strategy to merge individual features from labeled and unlabeled images. Unsupervised representation learning is to learn features from unlabeled data. Another significant character for unsupervised methods is they tend to learn low-dimensional features that captures dominant information of high-dimensional features. Common approaches in this group such as K-means clustering, principle component analysis (PCA), or deep/multilayer architecture neural networks. [43] took advantage of speed and scalability of the K-means approach and used it to extract image representations based on calculating Euclidean distance from image patch to learn centroids. Because K-means is extremely fast, no complicated hyperparameters to tune beyond the model structure and easy to implement, thus [28] applied K-means clustering to learn centroids from unlabled input data and chose two distance-based feature mappings given learned centroids to obtain feature representations. PCA is another commonly used unsupervised feature learning strategy, which is a linear feature learning approach with solid linear algebra foundation that is often used for dimensionality reduction of data. PCA produces k singular vectors corresponding to k largest singular values of data matrix of n unlabeled input data vectors, note k is much smaller than n, and these k singular vectors are feature vectors learned from the input data. [128] used a sparse PCA to select biomarkers. [30] proposed a hybrid technique based on PCA and facial feature extraction for frontal face detection in color images. There are several limitations for PCA, its assumption regarding that the directions with large variance are of most significance which might not be held in different applications. In addition, PCA only exploits first and second order of the original data, which may not well characterize the data distribution. A new trend in representation learning is to use deep architecture neural systems to learn "deep features". These multilayer neural network architecture are based on distributed representation [58] assumption which was first proposed by Geoffrey Hinton in 1984. Distributed representation is an extension of local representation which refers to each neuron in the neural network merely offers a local representation to certain concept, while a distributed representation means a many-to-many relationship between distinct types of representations. In distributed representation, each concept is represented by multiple neurons, each of which, on the other hand, participates in the representation of many concepts. In this paper we mainly focus on representative representation learning approaches which has gained much attention and achieved outstanding performance in learning discriminative feature vectors for text.

4 PROBABILISTIC REPRESENTATIONAL LEARNING

In section 3 we can observe neural network has become a very popular method to learn representation vectors especially in semi-supervised and unsupervised machine learning problems. Based on this algorithm distributed representation learning can be implemented to learn more accurate and abstract representations. On the other hand, another commonly used method in representation learning is based on building probabilistic models which will be introduced in this section.

4.1 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis (PLSA) can be treated as a representative method that stems from a statistical perspective of LSA for modeling co-occurrence data arising in natural language processing [37]. The significant difference from LSA, which relies on linear algebra and performs SVD on term-document matrix to derive low-dimensional representation for observed variables, is PLSA is based on a mixture decomposition model derived from a latent class model called the aspect model [61] that has sound statistical foundation. Expectation Maximization (EM) algorithm is the standard procedure used for fitting the latent variable model [62].

It is meaningful to discuss advantages and shortcomings of PLSA compared to LSA. T.Hofmann clarifies the relationship between those two algorithms. It points out the crucial difference between these two is the objective function, In LSA it is the Frobenius form, yet PLSA replies on the likelihood function. PLSA is superior to LSA on the modeling side because it has clear probabilistic meaning like conditional independence and well-defined probability distribution. Furthermore, directions in the LSA latent space cannot be interpreted directly, each direction in this PLSA latent space corresponds to class-conditional word distributions defining a specific topical context. In addition, PLSA applies probabilistic theory to fit and select model and control the model complexity. On the contrary, LSA can only choose dimension value on heuristics. However, LSA performs better than PLSA regarding computational complexity, because SVD can be computed exactly, while EM often suffers from the local maximum of likelihood function.

People have discovered advantages of PLSA and applied it on multiple tasks. [21] took advantage of the feature that PLSA can provide a better representation for sparse information in a text block to do topic-based document segmentation. [67] used PLSA to mining the hideen semantic associations between users and web pages based on co-occurrence patterns of pages in user sessions. [39] proposed a hybrid model of mixing Nonnegative matrix factorization (NMF) and PLSA to help PLSA jump out of the local optima. PLSA also be performed by Hofmann [63] to learn a collaborative filtering model so as to capture user preferences.

4.2 Topic Modeling

Probabilistic topic models are generative processes aimed at discovering the hidden thematic structure of large collections of documents. They do not require class labels or annotations making them unsupervised learning algorithms [17].

One of the simplest topic model was proposed by Blei et al. in 2003 called Latent Dirichlet Allocation (LDA) [19]. Using the words of the documents as the observed variables they define the hidden topic structure characterized by three components: topics, distribution of topics in a document and per-word per-document topic assignments. Being a generative process, a joint probability distribution is defined over both the observed and hidden variables and the conditional probability (posterior) of observing the hidden over the observed data is computed. The model assumes that the order of the words and documents is not important and the number of topics are assumed to be known and fixed. However, even though each document is projected on a fixed number of topics, each document can exhibit different proportions. The topic distribution and posterior are commonly used for dimensionality reduction, clustering, summarizing, inference and classification of text documents.

Over the years, numerous variations of LDA have been proposed by either relaxing the constraints or for suiting the application domain better by incorporating meta data (additional information about the documents). As discussed before, traditional LDA does not account for the order of the documents. However, some applications require studying the evolution of topics in a stream of documents [3, 47, 56, 68, 115, 151]. Blei el. al proposed a dynamic topic model to discover evolving set of topics in sequentially organized corpus of documents [18]. The method uses a state space model which divides the time into distinct periods. Wang et al. generalized the topic evolution from discrete time windows to a continuous space [138]. Another method to discover topics called Online Topic Model (OLDA) was proposed by AlSumait et al. where the topic model was incrementally updated with new set of documents [7]. Wallach et al. relaxed the constraint for word independence and presented a topic model where each word was conditioned on the previous word [136]. Similarly, Griffiths et al. explored different dependencies amongst words based on syntactic and semantic use. Their model uses HMM for syntax and LDA for semantic to better under the role of a word. The model produced competitive results for parts of speech and classification [51]. Further, since topic proportions are randomly drawn from the Dirichlet distribution, the topic proportions are considered near independent, however realistically it is possible that topic proportions are correlated to each other. For example, a machine learning research paper is more likely to have topics on neural networks and representation learning rather than sports. In order to address this drawback, Blei et al. proposed Correlated Topic Model (CTM) where topic proportions were correlated according to the logistic normal distribution [16]. As discussed earlier, LDA assumes a fixed number of topics however Teh et al. proposed the use of hierarchical Dirichlet process in order to relax this constraint [131]. Hoffman et al. proposed yet another extension for large document collections called online learning of lda [60] which was based on online stochastic optimization and estimated the topics much faster than traditional LDA.

In addition to the documents other attributes of the documents might be available which usually is called the meta data. Examples of meta data could be a response variable for a document, user tags and author information. The traditional LDA model is an unsupervised method. However, supervised LDA (sLDA) can be used when a response variable (categorical or continuous score) is available. It models the documents and the response variables together by maximizing the joint likelihood of the data and response variable. The response is dependent on the empirical topic proportions found in a document [90]. A variant of sLDA called medLDA learns the topic distributions using a max-margin discriminative method [152, 153]. Along the same line, DiscLDA aims to find topic proportions in the reduced dimension space conditioned on the response variable [73]. However, sLDA, medLDA, and DiscLDA only deal with one label associated with a document. Ramage et al. proposed Labeled LDA to discover topics in multi-labeled documents [110]. Partially LDA generalizes Labeled LDA by allowing multiple topics for one label [111].

Further, there are variations of the model that have been enriched with external information and used for multiple applications. We discuss few of the applications: Author-Topic Model aims at modeling the topic distribution for the content and the authors simultaneously [114]. Similarly, Topic-Link LDA models topics in documents leveraging author connections and their influence on each other [85]. Furthermore Wang et al., leveraged the latent themes obtained by probabilistic topic models to recommend scientific articles [139]. Titove et al. leveraged probabilistic topic models such as LDA and PLSA to infer rateable aspects from online user reviews by extracting fine grained topics [132]. Ramage et al. utilized Labeled LDA to summarize a user's Twitter stream [109]. In order to deal with short text used for social media sites such as Twitter, Nigam et al. developed a framework leveraging LDA to infer topical interests of followers of a company [100]. Zhao et al. contrasted Twitter as an information source to New York Times by using a topic model called Twitter-LDA model [150].

5 DISTRIBUTED REPRESENTATIONAL LEARNING

5.1 Character Level Neural Language Modeling

Even though word level Neural Lanugage Models (NLM) have been outperformed count based n-gram models, it is agnostic to the sub word information, such as morphemes, this can be problematic for morphologically rich languages[69]. Sutskever et al. [127] propose a new Multiplicative Recurrent Neural Network (RNN) to model the character sequence by taking advantage of the advance in Hessian-Free optimization. Graves et al.[50] use Long Short-Term Memory (LSTM) RNN to model complex sequences, such as character sequence.

Improvements have been reported for the part-of-speech tagger task[41] and named entity recognition task[117] employing a hybrid embedding scheme, which concatenates the word embedding with the character level embedding. Zhang et al.[149] take advantage of the deep Convolution Neural Networks (ConvNets) to use character vocabulary as input to learn a complete character level embedding for a text classification task, the experiment results show that the proposed model outperforms the pretrained

word2vec[91, 92] embedding based LSTM architectures on several evaluation data sets.

Ling et al.[84] use the one hot encoding of each character in the word as bidirectional LSTM RNN's input in forward and backward directions to train a character level embedding, they apply a similar architecture to the part-of-speech tagging task. Ballesteros et al.[10] use a similar architecture to [84], a bidirectional LSTM RNN over characters in both from left to right and from right to left directions to train a transition-based parser, the proposed method obtains improvements on many morphologically rich languages.

5.2 Word Embeddings

There has been a lot of work in the field of continuous vectors for words as previously discussed through other models. Turian et al. discussed various unsupervised methods to obtain word features before utilizing them in supervised approaches [134]. However, in this section, we focus on distributed representations of words learned by neural networks and their applications. Distributed word representations are called word embeddings [134].

Distributed representations have shown to outperform n-gram models [91–93]. Word2vec model proposed by Mikolov et al. was able to learn word vectors to capture syntactic and semantic similarities. Additionally, the vectors enabled us to perform algebraic operations in the continuous space. One of the most common examples used to demonstrate the model's effectiveness is $King - Man + Woman = Queen$. For efficient computation, two architectures were introduced: 1) Continuous bag of words model (CBOW) which predicts the current word based on the context 2) Continuous skip-gram model (SG) which maximizes classification of a word based on the context. Both the architectures leveraged the probability distribution over the next word enabling them to generalize better than traditional n-gram models [93]. Goldberg et al. provided a detailed discussion of the negative sampling to further understand word2vec model [48]. Representations from skip gram model with negative sampling (word2vec implementation) have been applied to various tasks such as named entity recognition [122], sentiment classification [143, 148]. word2vec for NER [122].

Mnih et al [96] presented a similar framework as Mikolov et al. however used variants of log-bilinear model to obtain speedup. Further, Pennington et al. argued that previous methods were able to capture local relationships however did not incorporate global relations. Therefore, they proposed GloVE which leveraged global matrix factorization and local context window to infer word representations [107]. The model used global log-bilinear regression models and has shown to perform well on many word similarity and named entity recognition tasks.

More recently, researchers have started looking at leveraging existing semantic lexicons to improve the learnt word representations. Bollegala et al. proposed a framework for joint word representation using co-occurrence as seen in existing embedding models and enriching them with relational constraints as observed from semantic lexicons. They reported high performance for word similarity on comparison to other models that leverage semantic lexicons [20]. Additionally, researchers are also looking at having multiple embedding for a given word [99].

Interestingly, Levy et al. compared four word representations: PPMI matrix, SVD factorization, skip gram model with negative sampling and Glove. According to their experiments, they found that the success of word embeddings can be attributed to the system design and hyper-parameters. On transferring this knowledge to traditional methods, there did not find a significant difference in the performance [80].

5.3 Phrase and Sentence Embeddings

An inherent limitation of word embeddings is they are blind to word order and inability to represent phrases. Learning vector representations for phrase was hence developed motivated by word embedding limitation. Some idiomatic phrases have a meaning that cannot be regarded as the simple composition of individual word meanings, for example, "Delta Airlines" is not a natural combination of meaning of words "Delta" and "Airlines". Extension from vector representations for words to entire phrases makes the Skip-gram model more expressive. To learn vector representation for phrases, people first need to identify reasonable phrases from text by finding out words which are frequently occur together yet infrequently in other contexts, and use token to represent a phrase in the training data. Mikolov el. used unigram and bigram counts based methods to form phrases [92]. Skip-gram model is trained in order to learn a vector representations for individual phrases. Mikolov et al. found the skip-gram model exhibits linear structure, based on which they proposed an additive compositionality approach as well to generate phrase representations by meaningfully combine words by element-wise addition of their vector representations. [130] treated learning a Twitter-specific sentiment classifier as a phrase-level sentiment classification task, and learned a sentiment-specific phrase embedding (SSPE), which are used as features for classification, from a neural network trained from large-scale tweets. [125] introduced a recursive neural network to learn continuous vector representations for phrases without manual engineering in order to capture full syntactic and semantic information and use them as input to parser to improve parser quality. There are also some works studying how to extend learning distributed representations for words or phrases to sentences. [124] attempted to learn sentence vector representations for a parse natural language sentences task by merging words representations into phrase representations in an order given by a parse tree of a sentence. [77] proposed the concept of "paragraph vector" which is capable of constructing vector representations for variable length of input sequences ranging from sentences, paragraphs to even documents. Kiros et al. proposed the skip-thought model [70] which casted learning sentence representation to train a encoder and decoder model, namely, for consecutive sentences S_1, S_2 and S_3, the skip-thought model can predict (decode) context sentences S_1 and S_3 given (encode) S_2. Some applications replying on sentence representation leaning can be found in recent years. [71] applied sentence embedding to the problem of predicting labels for sentences given labels for reviews. [35] utilized sentence vector representation on short-texts analysis in order to grasp semantic meaning of such texts.

5.4 Paragraph Embeddings

Two most common ways of representing paragraphs and documents are through BOW and TF-IDF representations of terms contained in the paragraphs or documents. Similar to word embeddings where each term or word has an exact and unique meaning represented by different weights in each element of the word embedding, a paragraph or document's meaning in paragraph document embeddings is represented by each element of its paragraph's or document's embedding, respectively. Each paragraph or document embedding is therefore represented by a combination of word embeddings of its containing words. Similar to how word embeddings represent different meanings of each word, paragraph and document embeddings represent different meanings of each paragraph and document. Consequently, paragraph and document vectors close to each other may therefore be of similar topics.

In their work on distributed representations of sentences and documents, [77] proposed Paragraph Vectors (PV) as an unsupervised method for learning continuous distributed vector representations of variable length texts ranging from sentences to documents. Two PV models were proposed: a Distributed Bag of Words (PV-DBOW) where sparse paragraph vectors are mapped into low-dimensional dense vectors and a Distributed Memory (PV-DM) model where the paragraph token acts as a memory token that remembers what is missing from the current context or the paragraph topic. The paragraph vector representations are learned to predict the surrounding words in the contexts sampled from the paragraph. In their study, the authors use paragraph vectors to learn embeddings of movie review texts that can be leveraged for sentiment analysis and Information Retrieval (IR) achieving an error rate of 7.42%. Compared to other approaches on the IMDB data set, their method is the only approach that goes significantly below the 10% error rate.

However, [77]'s PV-DBOW model was not designed for IR because its learning objective excessively suppresses the importance of frequent words, is prone to over-fitting short documents during training iterations, and fails to model word-context associations thereby making it difficult to capture word substitution relationships that are important in IR. Recently, [5] proposed three major improvements over the original PV model to adapt it to IR tasks: replacing the corpus frequency-based negative sampling strategy with a document frequency-based strategy, regularizing document representations to prevent over-fitting of short documents, and introducing a joint learning objective over document-word and word-context associations to enhance word probability estimation. The author's study on how to effectively use the PV model in language model frameworks to improve ad-hoc information retrieval demonstrated that a PV model can outperform topic models on language model estimation for IR.

Following [77]'s work, [33] considered tasks for PVs other than sentiment analysis and information retrieval. In their work on document embeddings with PVs, the authors compare PVs with other baselines such as LDA on two tasks: finding the nearest Wikipedia articles an audience should browse, given an initial article and finding related articles on arXiv. Their model jointly trains word embeddings with paragraph vectors to improve the quality of the paragraph vectors. The results of their experiment showed that paragraph vectors are superior to LDA for measuring semantic similarity on Wikipedia articles and on par with LDA's best performing number of topics on finding related arXiv articles. The authors further propose additional applications of paragraph vectors such as local and non-local corpus navigation, data set exploration, book recommendation, and reviewer allocation.

5.5 Document Emdeddings

In addition to paragraph embeddings, document-level representation is an essential pre-processing technique for reducing the complexity of documents and making them easier to handle in machine learning applications. The goal of document-level representations is to map documents into compact forms of their contents. This can be achieved by transforming the full text of each document into a document vector of term weights or word frequencies from a set of terms, also known as a dictionary, that occur at least once in a minimum number of documents.

Each document is therefore represented by a vector, called document embedding, calculated by the vector representation of its containing words. Due to the huge size of unique terms from the document corpus that can be contained in a dictionary, dimensionality reduction is often performed to eliminate irrelevant and redundant features that impact the performance of classification algorithms both in terms of running time and classification accuracy. Some common applications of document vector representations include document retrieval[9, 144], clustering [72], and classification [66, 72, 74, 144].

Similar to Earth Mover's Distance, Kusner et al. [72] developed a Word Mover's Distance (WMD) function to measure dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. Compared to other document distance measures, the WMD is hyper-parameter free, highly interpretable, and naturally incorporates knowledge encoded in the word2vec space. The WMD measure can be used to calculate document similarity for document retrieval, clustering and classification tasks. Such similarities can also be used for ranking and recommendation systems.

In information retrieval, [9] explore several document representation models for blog relevance ranking - the task of recommending blogs to a user in response to a query. The authors proposed two document representation models for blogs in IR: a Large Document (LD) model in which entire blogs are indexed as a single document and a Small Document (SD) model where indexing is performed at the blog post level and the final blog ranking is computed through its aggregate posts rankings. The experiments were conducted using the TREC Blog06 collection, exclusively focusing on feeds while ignoring permalinks and homepage documents. The feed documents were a combination of ATOM and RSS XML. Although both the LD and SD models perform at comparable levels of precision and recall, the results of the experiments demonstrated that the two models are complementary and combining the two models provides superior results to either model in isolation.

Huang et al. [66] proposed a method to learn document embeddings with neural networks for text classification tasks. The authors use a BOW representation of a document's words to derive

the document embeddings and compare the results of their neural network architecture with simple BOW vector representations for text classification. Using two data sets for evaluation, the 4th Large Scale Hierarchical Text Classification Challenge (LSHTC) and Sogou data sets, their results showed that document embeddings have a higher classification accuracy than BOW vectors.

Lai et al. [74] recently presented a framework for text classification using a recurrent convolutional neural network. In their framework, they incorporate the contextual information by defining a left and right context instead of using a window based approach. They compared their framework against traditional methods such as BOW and n-gram models. Further, the framework was benchmarked against state of the art methods such as LDA and para2vec. Their framework performed better for 3 of the 4 datasets used (20Newsgroup, Fudan, ACL Anthology Network and Stanford Sentiment Treebank).

In their research on learning document semantic representation with Hybrid Deep Belief Network (HDBN), Yan et al. [144] propose a high-level abstraction semantic representation method for document retrieval and classification. The authors proposed a new HDBN which uses Deep Boltzman Machine (DBM) on the lower layers together with Deep Belief Network (DBN) on the upper layers. Compared to other neural network architectures, the advantage of DBM in their model is that it employs undirected connections when training weight parameters which can be used to effectively sample the states of the nodes on each layer.

6 FUTURE OF TEXT REPRESENTATION

In this survey, we have introduced various algorithms that enable us to capture rich information in text data and represent them as vectors for traditional machine learning frameworks. We firstly discussed traditional methods of text representation which mostly involved feature engineering. The models such as BOW or semantic based were primarily count based. LSA took the representation one step further where it performed dimensionality reduction on a document-term matrix and subsequently computed latent features. The focus shifted from feature engineering to learning with probabilistic models such as PLSA and LDA. These models did not involve manual efforts for feature engineering but rather introduced unsupervised methods for obtaining a probabilistic thematic representation. More recently, with the advent of high computation power, neural networks are commonly used for learning representation of text at character, word, sentence, paragraph or document level.

Deep learning techniques have been attracting much attention in these years which are well known especially for their capability of addressing problems in computer vision and speech recognition areas. The great success deep learning achieved stems from its use multiple layers of nonlinear processing units for learning multiple layers of feature representations of data, different layers correspond to different abstraction levels. Yann LeCun et al. proposed a tight connection between deep learning and representation learning in [78], they considered deep learning methods as representation learning methods with multiple levels of representation, such representations are usually automatically obtained by composing non-linear modules that each transform the representation at one level into a representation at a higher, slightly more abstract level. Such multi-level architecture overcomes the weakness of utilizing conventional feature learning strategies in representation learning, because layers of features in deep learning are not designed by people mastering domain knowledge, they are directly learned from raw data. As shown in the paper, there is a growing trend to explore deeper models for previously popular shallow models. The deep models are achieving state of the art performance for various machine learning tasks [46]. In this section we will briefly discuss some representative works by introducing deep learning methods into text representation learning problems we discussed in previous sections.

The LSA strategy uses linear function for feature computation and are unsupervised, so does PLSA given that PLSA alternatively can be presented as a document-word matrix factorization approach yet in a probabilistic way. Those linear feature learning approaches usually show weakness in learning semantic representations. As Yoshua Bengio described in [14], the expressive power of linear features is very limited. Linear features can barely be stacked to form deeper, more abstract representations since the composition of linear operations yields another linear operation. In recent years, there are a number of literatures [40, 116, 140] had proposed efficient methods applying deep learning techniques to design novel representation learning methods that are exploited to expand the conventional semantic indexing on text data, which achieved outstanding results that outperform traditional methods dramatically. Essentially, deep learning methods are formed by the composition of multiple non-linear transformations that will generate more compact and useful representations.

As previously discussed distributed representations have helped address the problem of curse of dimensionality and increase generalization. Salakhutdinov et al. [59] proposed Replicated Softmax, simple two-layer undirected topic model, to infer distributed representation of documents. They showed that their model generalized better than probabilistic topical models such as LDA. Inspired from Replicated Softmax, Larochelle et al. [76] proposed a neural network topic model where they used neural autoregressive distribution estimator as an alternative to RBMs. Gan et al. [46] proposed a deep generative model for topic modelling combining traditional Bayesian approach and sigmoid deep belief network.

Deep learning methods not only shows powerful capability in semantic analysis applications on text data, but can be successfully used in number of tasks of text classification and natural language processing. Character level embedding methods [69, 149] demonstrate success in modeling short text, such as tweets and reviews, which can capture emotions or polarities expressed by multiple punctuations and word variations. Representative works [1, 83, 86, 105] by introducing deep neural network models to learn phrase or sentence embeddings demonstrated satisfying results.

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207* (2016).

[2] Charu C Aggarwal and ChengXiang Zhai. 2012. *Mining text data*. Springer Science & Business Media.

[3] Amr Ahmed and Eric P Xing. 2012. Timeline: A dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. *arXiv preprint arXiv:1203.3463* (2012).

[4] Mohammad Salim Ahmed and Latifur Khan. 2009. Sisc: A text classification approach using semi supervised subspace clustering. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 1–6.

[5] Qingyao Ai, Liu Yang, Jiafeng Guo, and W Bruce Croft. 2016. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 869–872.

[6] Cyril N Alberga. 1967. String similarity and misspellings. *Commun. ACM* 10, 5 (1967), 302–313.

[7] Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. 2008. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 3–12.

[8] Richard C Angell, George E Freund, and Peter Willett. 1983. Automatic spelling correction using a trigram similarity measure. *Information Processing & Management* 19, 4 (1983), 255–261.

[9] Jaime Arguello, Jonathan L Elsas, Jamie Callan, and Jaime G Carbonell. 2008. Document Representation and Query Expansion Models for Blog Recommendation. *ICWSM* 2008, 0 (2008), 1.

[10] Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. *arXiv preprint arXiv:1508.00657* (2015).

[11] Ershad Banijamali and Ali Ghodsi. 2016. Semi-Supervised Representation Learning based on Probabilistic Labeling. *arXiv preprint arXiv:1605.03072* (2016).

[12] Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. *Advances in automatic text summarization* (1999), 111–121.

[13] Florian Beil, Martin Ester, and Xiaowei Xu. 2002. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 436–442.

[14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[15] Michael W Berry and Malu Castellanos. 2004. Survey of text mining. *Computing Reviews* 45, 9 (2004), 548.

[16] David Blei and John Lafferty. 2006. Correlated topic models. *Advances in neural information processing systems* 18 (2006), 147.

[17] David M Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (2012), 77–84.

[18] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 113–120.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[20] Danushka Bollegala, Alsuhaibani Mohammed, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Joint word representation learning using a corpus and a semantic lexicon. *arXiv preprint arXiv:1511.06438* (2015).

[21] Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. 2002. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 211–218.

[22] Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Citeseer.

[23] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–479.

[24] Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32, 1 (2006), 13–47.

[25] Xavier Carreras and Lluis Marquez. 2001. Boosting trees for anti-spam email filtering. *arXiv preprint cs/0109015* (2001).

[26] William B Cavnar. 1993. N-gram-based text filtering for TREC-2. *Ann Arbor* 1001 (1993), 48113–4001.

[27] Minmin Chen, Kilian Q Weinberger, Fei Sha, and others. 2013. An alternative text representation to TF-IDF and Bag-of-Words. *arXiv preprint arXiv:1301.6770* (2013).

[28] Adam Coates and Andrew Y Ng. 2012. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*. Springer, 561–580.

[29] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political Polarization on Twitter. *ICWSM* 133 (2011), 89–96.

[30] Saman Cooray and Noel OfiConnor. 2004. Facial feature extraction and principal component analysis for face detection in color images. In *International Conference Image Analysis and Recognition*. Springer, 741–749.

[31] Gordon V Cormack, José María Gómez Hidalgo, and Enrique Puertas Sánz. 2007. Feature engineering for mobile (SMS) spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 871–872.

[32] Ronald W Cornew. 1968. A statistical method of spelling correction. *Information and Control* 12, 2 (1968), 79–93.

[33] Andrew M Dai, Christopher Olah, and Quoc V Le. 2015. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998* (2015).

[34] Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7, 3 (1964), 171–176.

[35] Cedric De Boom, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. 2016. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters* 80 (2016), 150–156.

[36] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391.

[37] Karthik Devarajan, Guoli Wang, and Nader Ebrahimi. 2015. A unified statistical approach to non-negative matrix factorization and probabilistic latent semantic indexing. *Machine learning* 99, 1 (2015), 137–163.

[38] Lipika Dey and SK Mirajul Haque. 2009. Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)* 12, 3 (2009), 205–226.

[39] Chris Ding, Tao Li, and Wei Peng. 2006. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *AAAI*, Vol. 6. 137–143.

[40] Cícero Nogueira Dos Santos and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts.. In *COLING*. 69–78.

[41] Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning Character-level Representations for Part-of-Speech Tagging.. In *ICML*. 1818–1826.

[42] Susan T Dumais. 2004. Latent semantic analysis. *Annual review of information science and technology* 38, 1 (2004), 188–230.

[43] Murat Dundar, Qiang Kou, Baichuan Zhang, Yicheng He, and Bartek Rajwa. 2015. Simplicity of Kmeans versus Deepness of Deep Learning: A Case of Unsupervised Feature Learning with Limited Data. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*. IEEE, 883–888.

[44] Ronen Feldman and James Sanger. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

[45] Dario Figueira, Loris Bazzani, Ha Quang Minh, Marco Cristani, Alexandre Bernardino, and Vittorio Murino. 2013. Semi-supervised multi-feature learning for person re-identification. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*. IEEE, 111–116.

[46] Zhe Gan, Changyou Chen, Ricardo Henao, David E Carlson, and Lawrence Carin. 2015. Scalable Deep Poisson Factor Analysis for Topic Modeling.. In *ICML*. 1823–1832.

[47] Andre Gohr, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. 2009. Topic evolution in a stream of documents. In *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 859–870.

[48] Yoav Goldberg and Omer Levy. 2014. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[49] Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 19–25.

[50] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[51] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2004. Integrating Topics and Syntax.. In *NIPS*, Vol. 4. 537–544.

[52] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[53] Vishal Gupta, Gurpreet S Lehal, and others. 2009. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence* 1, 1 (2009), 60–76.

[54] Khaled M Hammouda and Mohamed S Kamel. 2004. Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on knowledge and data engineering* 16, 10 (2004), 1279–1296.

[55] Allen R Hanson, Edward M Riseman, and E Fisher. 1976. Context in word recognition. *Pattern Recognition* 8, 1 (1976), 35–45.

[56] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. 2009. Detecting topic evolution in scientific literature: How can citations help?. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 957–966.

[57] Marti A Hearst. 1999. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 3–10.

[58] Geoffrey E Hinton. 1984. Distributed representations. (1984).

[59] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*. 1607–1614.

[60] Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*. 856–864.

[61] Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 289–296.

[62] Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning* 42, 1-2 (2001), 177–196.

[63] Thomas Hofmann. 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 259–266.

[64] Xia Hu and Huan Liu. 2012. *Text Analytics in Social Media*. Springer US, Boston, MA, 385–414. DOI: http://dx.doi.org/10.1007/978-1-4614-3223-4_12

[65] Anna Huang. 2008. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. 49–56.

[66] Chaochao Huang, Xipeng Qiu, and Xuanjing Huang. 2014. Text classification with document embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 131–140.

[67] Xin Jin, Yanzan Zhou, and Bamshad Mobasher. 2004. Web usage mining based on probabilistic latent semantic analysis. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 197–205.

[68] Yookyung Jo, John E Hopcroft, and Carl Lagoze. 2011. The web of topics: discovering the topology of topic evolution in a corpus. In *Proceedings of the 20th international conference on World wide web*. ACM, 257–266.

[69] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615* (2015).

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

[71] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 597–606.

[72] Matt J Kusner, Yu Sun, Nicholas I Kolkin, Kilian Q Weinberger, and others. 2015. From Word Embeddings To Document Distances.. In *ICML*, Vol. 15. 957–966.

[73] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. 2009. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Advances in neural information processing systems*. 897–904.

[74] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In *AAAI*, Vol. 333. 2267–2273.

[75] Thomas K Landauer. 2006. *Latent semantic analysis*. Wiley Online Library.

[76] Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*. 2708–2716.

[77] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[78] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[79] Joo Ho Lee and Jeong Soo Ahn. 1996. Using n-grams for Korean text retrieval. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 216–224.

[80] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics* 3 (2015), 211–225. https://www.transacl.org/ojs/index.php/tacl/article/view/570

[81] Yanjun Li, Soon M Chung, and John D Holt. 2008. Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering* 64, 1 (2008), 381–404.

[82] Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[83] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *arXiv preprint arXiv:1703.03130* (2017).

[84] Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096* (2015).

[85] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. 2009. Topic-link LDA: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*. ACM, 665–672.

[86] Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. 2016. Sentence Ordering using Recurrent Neural Networks. *arXiv preprint arXiv:1611.02654* (2016).

[87] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. 2009. Supervised dictionary learning. In *Advances in neural information processing systems*. 1033–1040.

[88] Inderjeet Mani and Mark T Maybury. 1999. *Advances in automatic text summarization*. Vol. 293. MIT Press.

[89] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, and others. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge university press Cambridge. 403–417 pages.

[90] Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. In *Advances in neural information processing systems*. 121–128.

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

[92] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[93] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations.. In *Hlt-naacl*, Vol. 13. 746–751.

[94] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[95] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography* 3, 4 (1990), 235–244.

[96] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2265–2273. http://papers.nips.cc/paper/5165-learning-word-embeddings-efficiently-with-noise-contrastive-estimation.pdf

[97] Robert Morris and Lorinda L Cherry. 1975. Computer detection of typographical errors. *IEEE Transactions on Professional Communication* 1 (1975), 54–56.

[98] Suleiman H Mustafa and Qasem A Al-Radaideh. 2004. Using N-grams for Arabic text searching. *Journal of the Association for Information Science and Technology* 55, 11 (2004), 1002–1007.

[99] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654* (2015).

[100] Aastha Nigam, Salvador Aguinaga, and Nitesh V Chawla. 2016. Connecting the Dots to Infer Followersfi Topical Interest on Twitter. In *Behavioral, Economic and Socio-cultural Computing (BESC), 2016 International Conference on*. IEEE, 1–6.

[101] Kamal Nigam, Andrew McCallum, and Tom Mitchell. 2006. Semi-supervised text classification using EM. *Semi-Supervised Learning* (2006), 33–56.

[102] Robert Nussbaum and Hans-Jörg Schek. 1978. *Automatic error detection in natural language words*. Heidelberg Scientific Center.

[103] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM* 11, 122-129 (2010), 1–2.

[104] Olumide Owolabi. 1993. Efficient pattern searching over large dictionaries. *Information processing letters* 47, 1 (1993), 17–21.

[105] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24, 4 (2016), 694–707.

[106] Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the eacl 2006 workshop making sense of sense-bringing computational linguistics and psycholinguistics together*, Vol. 1501. Trento, 1–8.

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

[108] Denys Poshyvanyk, Andrian Marcus, Vaclav Rajlich, Y-G Gueheneuc, and Giuliano Antoniol. 2006. Combining probabilistic ranking and latent semantic indexing for feature identification. In *Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on*. IEEE, 137–148.

[109] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. 2010. Characterizing microblogs with topic models. *ICWSM* 10 (2010), 1–1.

[110] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 248–256.

[111] Daniel Ramage, Christopher D. Manning, and Susan Dumais. 2011. Partially Labeled Topic Models for Interpretable Text Mining. (August 2011). https://www.microsoft.com/en-us/research/publication/putting-search-into-context-and-context-into-search/

[112] Marc'Aurelio Ranzato and Martin Szummer. 2008. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the*

25th international conference on Machine learning. ACM, 792–799.

[113] Edward M Riseman and Roger W Ehrich. 1971. Contextual word recognition using binary digrams. *IEEE Trans. Comput.* 100, 4 (1971), 397–403.

[114] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 487–494.

[115] Ankan Saha and Vikas Sindhwani. 2012. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 693–702.

[116] Ruslan Salakhutdinov and Geoffrey Hinton. 2007. Semantic hashing. *RBM* 500, 3 (2007), 500.

[117] Cicero Nogueira dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008* (2015).

[118] Sam Scott and Stan Matwin. 1998. Text classification using WordNet hypernyms. In *Use of WordNet in natural language processing systems: Proceedings of the conference*. 38–44.

[119] Sam Scott and Stan Matwin. 1999. Feature engineering for text classification. In *ICML*, Vol. 99. 379–388.

[120] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34, 1 (2002), 1–47.

[121] Julian Sedding and Dimitar Kazakov. 2004. WordNet-based text document clustering. In *proceedings of the 3rd workshop on robust methods in analysis of natural language data*. Association for Computational Linguistics, 104–113.

[122] Scharolta Katharina Sienčnik. 2015. Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*. Linköping University Electronic Press, 239–243.

[123] Pawel Sobkowicz, Michael Kaschesky, and Guillaume Bouchard. 2012. Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web. *Government Information Quarterly* 29, 4 (2012), 470–479.

[124] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 129–136.

[125] Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*. 1–9.

[126] L Venkata Subramaniam, Shourya Roy, Tanveer A Faruquie, and Sumit Negi. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*. ACM, 115–122.

[127] Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1017–1024.

[128] YH Taguchi and Yoshiki Murakami. 2013. Principal component analysis based feature extraction approach to identify circulating microRNA biomarkers. *PloS one* 8, 6 (2013), e66714.

[129] Ah-Hwee Tan and others. 1999. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, Vol. 8. 65–70.

[130] Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach.. In *COLING*. 172–182.

[131] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2004. Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes.. In *NIPS*. 1385–1392.

[132] Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 111–120.

[133] Ivan Titov and Ryan T McDonald. 2008. A Joint Model of Text and Aspect Ratings for Sentiment Summarization.. In *ACL*, Vol. 8. Citeseer, 308–316.

[134] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 384–394.

[135] Julian R. Ullmann. 1977. A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. *Comput. J.* 20, 2 (1977), 141–147.

[136] Hanna M Wallach. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 977–984.

[137] Bin Wang and Wen-feng PAN. 2005. A survey of content-based anti-spam email filtering [j]. *Journal of Chinese Information Processing* 5, 000 (2005).

[138] Chong Wang, David Blei, and David Heckerman. 2012. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298* (2012).

[139] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[140] Hao Wu, Martin Renqiang Min, and Bing Bai. 2014. Deep Semantic Embedding.. In *SMIR@ SIGIR*. 46–52.

[141] Min Xiao and Yuhong Guo. 2013. Semi-Supervised Representation Learning for Cross-Lingual Text Classification.. In *EMNLP*. 1465–1475.

[142] Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 267–273.

[143] Bai Xue, Chen Fu, and Zhan Shaobin. 2014. A study on sentiment computing and classification of sina weibo with word2vec. In *Big Data (BigData Congress), 2014 IEEE International Congress on*. IEEE, 358–363.

[144] Yan Yan, Xu-Cheng Yin, Sujian Li, Mingyuan Yang, and Hong-Wei Hao. 2015. Learning document semantic representation with hybrid deep belief network. *Computational intelligence and neuroscience* 2015 (2015), 28.

[145] Yiming Yang. 1995. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 256–263.

[146] EM Zamora, Joseph J Pollock, and Antonio Zamora. 1981. The use of trigram analysis for spelling error detection. *Information Processing & Management* 17, 6 (1981), 305–316.

[147] Sarah Zelikovitz and Haym Hirsh. 2001. Using LSI for text classification in the presence of background text. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 113–118.

[148] Dongwen Zhang, Hua Xu, Zengcai Su, and Yunfeng Xu. 2015. Chinese comments sentiment classification based on word2vec and SVM perf. *Expert Systems with Applications* 42, 4 (2015), 1857–1863.

[149] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. 649–657.

[150] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*. Springer, 338–349.

[151] Ding Zhou, Xiang Ji, Hongyuan Zha, and C Lee Giles. 2006. Topic evolution and social interactions: how authors effect research. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 248–257.

[152] Jun Zhu, Amr Ahmed, and Eric P Xing. 2009. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 1257–1264.

[153] Jun Zhu, Amr Ahmed, and Eric P Xing. 2012. MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research* 13, Aug (2012), 2237–2278.