International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-020-01069-8

ORIGINAL ARTICLE

From static to dynamic word representations: a survey

Yuxuan Wang[1] · **Yutai Hou**[1] · **Wanxiang Che**[1] · **Ting Liu**[1]

Received: 15 September 2019 / Accepted: 16 January 2020

Abstract
In the history of natural language processing (NLP) development, the representation of words has always been a significant research topic. In this survey, we provide a comprehensive typology of word representation models from a novel perspective that the development from static to dynamic embeddings can effectively address the polysemy problem, which has been a great challenge in this field. Then the survey covers the main evaluation metrics and applications of these word embeddings. And, we further discuss the development of word embeddings from static to dynamic in cross-lingual scenario. Finally, we point out some open issues and future works.

Keywords Word representation · Static embedding · Dynamic embedding · Cross-lingual embedding

1 Introduction

In the history of NLP, how to represent words which are often regarded as the smallest semantic elements in the natural language has always been a research hotspot. In recent years, low-dimensional word representation vectors trained with massive amounts of unannotated textual data, so-called word embeddings [74, 86] have been demonstrated to be effective in a wide range of NLP tasks including POS tagging [119], syntactic parsing [13], named entity recognition [59] and semantic role labeling [129], machine translation [130]. This kind of embeddings is *static* in the sense that they do not change with the context once been learned. Despite their efficiency, the static nature of these embeddings makes it difficult to cope with the *polysemy problem*, since the meaning of a polysemous word depends on its context.

To deal with this problem, a number of approaches have been recently proposed to learn the representation of words among their contexts.[1] For example, in two sentences: "Apple sells phones" and "I eat an apple", dynamic embeddings will represent "apple" differently according to the contexts, while *static* embedding can not distinguish the semantic difference between two "apples". These dynamic embeddings extracted from pre-trained language models [26, 71, 87, 90] have demonstrated dramatic superiority over their static predecessors in various NLP tasks. In this survey, we will provide an inclusive overview of the existing static and dynamic embedding models. One of the main goals of this survey is to display the developing trend of word embedding models from static to dynamic, which demands for coping with the polysemy problem. To facilitate this, we first introduce existing static embedding models and some attempts to deal with the polysemy problem with them in Sect. 2. Then, we introduce and compare the recently proposed dynamic embedding models, and show how they are able to alleviate the polysemy problem in Sect. 3.

With the embedding models described, how to evaluate and apply them to downstream tasks are also notable. Therefore, we discuss two major categories of evaluation metrics, namely intrinsic and extrinsic ones in Sect. 4. Then we briefly introduce the line of works trying to transform the monolingual static and dynamic embeddings to cross-lingual scenarios in Sect. 5. Finally, we discuss some open issues of word representation in Sect. 6. Notions used in the paper are listed in (Table 1).

✉ Wanxiang Che
car@ir.hit.edu.cn

Yuxuan Wang
yxwang@ir.hit.edu.cn

Yutai Hou
ythou@ir.hit.edu.cn

Ting Liu
tliu@ir.hit.edu.cn

[1] Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin, China

[1] These embeddings are *contextualized* or *dynamic* as opposed to the traditional ones.

Published online: 17 February 2020

Table 1 Notions used through the paper

Notion	Description
w	A single word
V	Vocabulary of words
$\boldsymbol{C}$	An embedding matrix
$\boldsymbol{C}(w)$	Fetch embedding of word w from matrix $\mathbf{C}$
D	Corpus
L	Objective function
σ	Softmax function
$\boldsymbol{W}$	Uppercase-bold-italic symbol denotes matrix
$\boldsymbol{b}$	Lowercase-bold-italic symbol denotes vector

2 Static representation

The development of static word representations can be roughly divided into two stages. At the first stage, sparse and high-dimensional vectors were used to represent words. This kind of embeddings suffer from the problem of data sparsity and their high dimensionality, usually as large as the vocabulary size, also makes it hard to use them. To cope with these problems, at the second stage, dense and low-dimensional vectors were trained with large textual data to take the place of them. In this section, we first introduce word representation models presented in both stages, and then describe the polysemy problem as well as several works trying to solve it with the static embeddings.

2.1 One-hot and distributional representations

In the early age of natural language processing, words are represented with high-dimensional zero–one vectors, or so-called *one-hot word vectors*, in which all entries are zero except the single entry corresponding to the word, which is one. With this approach, all vectors are orthogonal to each other. Therefore, it is intractable to identify the semantic distance between words. For instance, the words *apple*, *orange* and *book* are equally similar to each other with the one-hot vector.

In order to model the syntactic and semantic similarity between words, additional features are leveraged to represent words, including: morphology (suffix, prefix), part-of-speech tags, dictionary features, such as word sense from WordNet[2], Brown word clustering [11]. Further, new methods were proposed under the distributional semantic hypothesis: *you shall know a word by the company it keeps* [33]. Here, a word is represented by its context, or specifically by a vector whose entries are the count of words that appear in context, which makes it possible to identify words that are semantically similar to each other. As an instance, by observing a large number of text corpora, we can find that the contexts of *apple* is much more similar to those of *orange* than *book*.

Formally, we denote by V_w the vocabulary of words and by V_c the vocabulary of predefined context words (Hence $|V_w|$ and $|V_c|$ are the vocabulary sizes of words and contexts respectively). Both of them are indexed, where w_i stands for the ith word in the word vocabulary and c_j the jth word in the context vocabulary. A matrix $\boldsymbol{W} \in \mathbb{R}^{|V_w| \times |V_c|}$ is used to quantify the correlation of words and their contexts, where

$$\boldsymbol{W}_{i,j} = n(w_i, c_j),$$

represents the correlation between word w_i and context c_j and $n(w_i, c_j)$ denotes the number of times c_j occurs in the context of w_i in a corpus D. And the size of D is denoted by $|D| = \sum_{w \in V_w, c \in V_c} n(w, c)$.

With such distributional representation, the semantic similarity between words can be easily quantified by measuring their distance in vector space, such as their cosine similarity or Euclidean distance. Therefore, we can say that the distributional representation has provided access to obtain the semantic similarity between words.

However, it is not always appropriate to measure the correlation of words solely by their co-occurrence, since overly high weights might be singed to word-context pairs containing common contexts. Consider the situation with the word *apple*, word-context pairs such as *an apple* and *the apple* would be observed much more frequently than *red apple* and *apple tree*, even though the latter ones are more informative.

An intuitive solution to this problem is by applying weighting factors such as *tf-idf*, which reduces the weights of word-context pairs proportionally to their frequency in the corpus [51], so that informative pairs receive relatively high weights. Whereas an alternative approach is quantifying the correlation of words and contexts with the *pointwise mutual information* (PMI) metric [23, 111], which measures the association between a pair of discrete outcomes x and y, defined as:

$$\mathrm{PMI}(x, y) = \log \frac{P(x, y)}{P(x)P(y)}.$$

In this case, the association between a word w and a context c is measured by $\mathrm{PMI}(w, c)$, which can be estimated with the actual observed numbers in the corpus. Hence the correlation between word w_i and context c_j in the word-context matrix changes to:

$$\boldsymbol{W}_{i,j} = \mathrm{PMI}(w_i, c_j) = \log \frac{n(w_i, c_j) \cdot |D|}{n(w_i) n(c_j)},$$

where $n(w_i) = \sum_{c \in V_c} n(w_i, c)$ is the frequency of w_i in the corpus D and $n(c_j) = \sum_{w \in V_w} n(w, c_j)$ is that of c_j in D.

[2] https://wordnet.princeton.edu/.

Another problem lying in the distributional representation is data sparsity, which is caused by the limitation of data. Some of the entries in the distributional vector may be incorrect since they are not observed in the limited data. Furthermore, the high dimension of the vectors, which depends on the predefined context vocabulary size, usually hundreds of thousands, is also bothersome.

To cope with these problems, dimensionality reduction mechanisms such as *Singular Value Decomposition* (SVD) [25] and *Latent Dirichlet Allocation* (LDA) [8] have been applied to condense the high-dimensional and sparse vectors to low-dimensional and dense ones. Take SVD as an example, which works by factorizing the matrix $\boldsymbol{W}$ into the product of three matrices $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{|V_w| \times |V_w|}$ is an unitary matrix, $\boldsymbol{\Sigma} \in \mathbb{R}^{|V_w| \times |V_c|}$ is a diagonal matrix of singular values, and $\boldsymbol{V} \in \mathbb{R}^{|V_c| \times |V_c|}$ is an unitary matrix. Note that the rank r of $\boldsymbol{\Sigma}$ is the same as that of $\boldsymbol{W}$. Firstly, we choose the top k ($k < r$) singular values from $\boldsymbol{\Sigma}$ to form a diagonal matrix $\boldsymbol{\Sigma}_k$, and let $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ be the matrices produced by selecting the first k columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively. Then their production $\boldsymbol{W}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$ can be regarded as a low rank approximation of the original matrix $\boldsymbol{W}$. According to the *Eckart-Young theorem* [28], $\boldsymbol{W}_k$ is the best *rank-k approximation* of $\boldsymbol{W}$ under L_2 loss.

The original matrix $\boldsymbol{W}$ is sparse, which is caused by insufficient data, in other words, with more text, $\boldsymbol{W}$ would have fewer zeros. Whereas the reconstructed matrix $\boldsymbol{W}_k$ is dense. From this point of view, SVD is able to simulate the unobserved text, and therefore can alleviate the data sparsity problem. Furthermore, the matrix $\boldsymbol{U}_k$ makes it possible to represent words with dense k-dimensional vectors instead of sparse $|V_c|$-dimensional ones (the value of k is typically hundreds, which is much smaller than $|V_c|$), which properly solved the high-dimensionality problem.[3]

2.2 Distributed representation

Distributional representation models introduced above have merits of easy implementation and good interpretability. But in most situations, they are slow to train and it is difficult to add new corpora. Besides, they lack ability to represent semantic elements of larger granularity, such as phrases and sentences. Therefore, a great amount of works have been proposed to directly learn low-dimensional, dense and continuous vectors (or so-called *distributed representations* or *word embeddings*) to represent words following the pioneering work *Neural Network Language Model* (NNLM) [7], where such vectors are learned implicitly with the language modeling task in a neural network architecture. A number of works in the beginning stage, like NNLM, use the distributed representation as a component of language models, therefore it is only regarded as a by-product. Given the breadth of this line of work, we only present a detailed introduction to the most prominent and influential approaches. For the works that have not received sufficient coverage, we provide pointers for readers interested in learning more details of them.

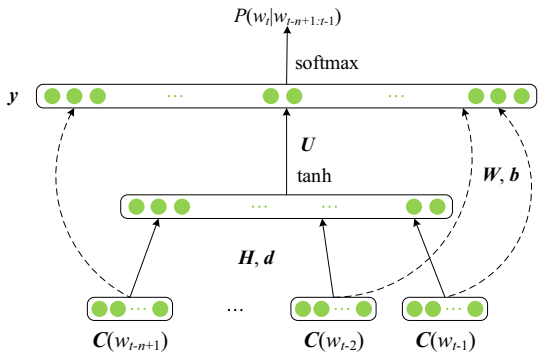

Fig. 1 Neural network architecture of NNLM, where $\boldsymbol{C}(w)$ is the distributed representation (vector) of word w

2.2.1 Neural Network Language Model

The *Neural Network Language Model* (NNLM) [7] is a pioneering work which introduces the idea of deep learning into language modeling and successfully mitigates the *curse of dimensionality* (i.e. Sequences in the test set is likely to have not been observed in the training data) by learning a distributed representation of words. The goal of language modeling is to learn a model that predicts the next word given previous ones. Practically, we assume the n-order Markov properties [70], and only the last $n-1$ words are considered when computing the probability distribution of the next word:

$$P(w_t|w_{1:t-1}) \approx P(w_t|w_{t-n+1:t-1}),$$

where $w_{i:j} = (w_i, w_{i+1}, \ldots, w_{j-1}, w_j)$. With the vocabulary of the training set denoted by V and its length by $|V|$, the model uses a matrix $\boldsymbol{C} \in \mathbb{R}^{|V| \times d}$ as a map where each of its row represents the *distributed representation (vector)* of word w in V, denoted by $\boldsymbol{C}(w) \in \mathbb{R}^d$.[4]

As depicted in Fig. 1, distributed vectors of previous $n-1$ words are concatenated and then fed into a feed-forward network, where the unnormalized log-probabilities $\boldsymbol{y} \in \mathbb{R}^{|V|}$ of output words are computed:

$$\boldsymbol{y} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{x} + \boldsymbol{U}\tanh(\boldsymbol{d} + \boldsymbol{H}\boldsymbol{x}).$$

[3] Please refer to [27] for more detailed comparison and analysis of these distributional representation models.

[4] We will use $\boldsymbol{C}(w)$ to denote the distributed embedding of word w in the rest of this paper.

Note that $\boldsymbol{x} = (\boldsymbol{C}(w_{t-1}); \boldsymbol{C}(w_{t-2}); \ldots; \boldsymbol{C}(w_{t-n+1}))$ is the concatenation of input distributed vectors. Then the probability of each output word w_t is computed with a *softmax function*:

$$P(w_t|w_{t-n+1:t-1}) = \frac{\exp(\boldsymbol{y}_t)}{\sum_i \exp(\boldsymbol{y}_i)}.$$

The model is trained by maximizing the following penalized objective function:

$$L = \frac{1}{T}\sum_t \log P(w_t|w_{t-n+1:t-1};\theta) + \lambda \cdot R(\theta),$$

where T is the number of training samples, θ is the overall parameters (including those of the neural network and the embedding matrix C) and $R(\theta)$ is a regularization term with weight λ.

Following the idea of using distributed representations from NNLM, a *Log-Bilinear Language Model* [79] was proposed, where a bilinear function is used to compute the probability of the next word instead of the feed-forward network in NNLM. More formally, the unnormalized probability is computed as follows:

$$P(w_t|w_{t-n+1:t-1}) = \sigma(\boldsymbol{C}(w_t)^\top \sum_{i=t-n+1}^{t-1} \boldsymbol{W}_i \boldsymbol{C}(w_i)),$$

where $\boldsymbol{W}_i \in \mathbb{R}^{d\times d}$ specifies the interaction between $\boldsymbol{C}(w_i)$ and $\boldsymbol{C}(w_t)$, σ is the softmax function, and biases are omitted for simplicity. Later the *Hierarchical Log-Bilinear Model* (HLBL) [80] was proposed based on the Log-Bilinear Language Model to speed-up its predicting stage.

A major deficiency of NNLM is that the feed-forward network can only observe a fixed length of context, which hinders it from exploiting longer context. Therefore, Recurrent Neural Networks (RNNs) are then used to replace the feed-forward network in the *Recurrent Neural Network Language Model* (RNNLM), which effectively reduces the perplexity [73].

2.2.2 SENNA

In all the works introduced above, distributed representations are only regarded as by-products of language models. Aiming at making use of the strong capability of such embeddings to facilitate more NLP tasks, a *Semantic Extraction using a Neural Network Architecture* (SENNA) system was proposed [20], where distributed embeddings are trained in a language model with a large amount of unlabeled data, and then used as input to downstream tasks including POS tagging, chunking, *Named Entity Recognition* (NER) and *Semantic Role Labeling* (SRL).

As depicted in Figure 2, SENNA makes use of a neural network to train the language model which is similar to

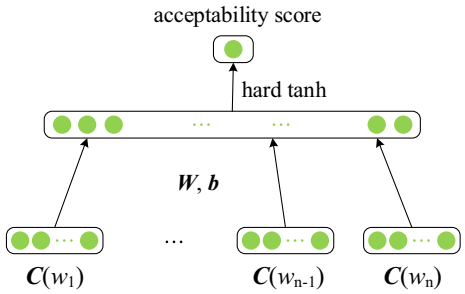

Fig. 2 Neural network architecture of SENNA

NNLM. But instead of estimating the *probability* of a word given the previous words, it computes scores describing the *acceptability* of a piece of text. This is because computing the normalization term of the probability with large dictionary size is extremely demanding, and sophisticated approximations are required. Therefore, in SENNA they instead use a pairwise ranking approach, seeking a network that assigns higher scores to legal phrases than to incorrect ones.

Formally, let $f_\theta(c)$ be the output score given a window of text $c = (w_1, w_2, \ldots, w_n)$, then the language model is trained by minimizing a hinge loss with respect to parameters θ in the network:

$$L = \sum_{c\in\mathcal{T}}\sum_{w\in\mathcal{D}} \max\{0, 1 - f_\theta(c) + f_\theta(c^{(w)})\},$$

where $\mathcal{T}$ is the set of all possible text windows with n words, $\mathcal{D}$ is the word dictionary, and $c^{(w)}$ represents a text window obtained by randomly replacing the central word of c with word w.

Practically, n was set to 11 in the paper, and it took 7 weeks to train the embeddings on a large amount of unlabeled data. These embeddings are then used as input to four supervised tasks (i.e. POS tagging, chunking, NER and SRL) in a similar network, where significant improvements have been observed.

2.2.3 CBOW and Skip-gram

SENNA innovatively applied the distributed representation to a bunch of NLP tasks other than language modeling, and demonstrated its capability of capturing textual information from unlabeled data to facilitate downstream tasks. However, the training of it is extremely time-consuming. Later, two prominent architectures, namely *Continuous Bag-of-Words* (CBOW) model and *Skip-gram* model [74], were proposed where the computational complexity is substantially reduced by simplifying the network architecture.

Concretely, the CBOW model predicts a word given its context. Fig. 3 shows its architecture, which is similar to

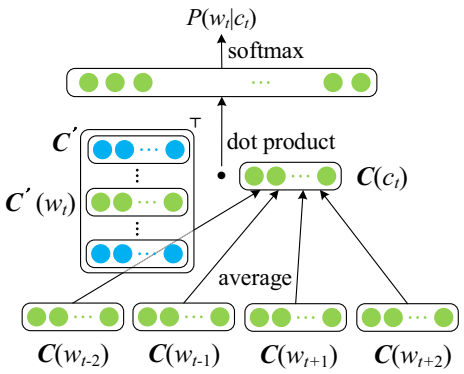

Fig. 3 Neural network architecture of CBOW model, where $\boldsymbol{C'}$ is the output embedding matrix containing vectors for central words

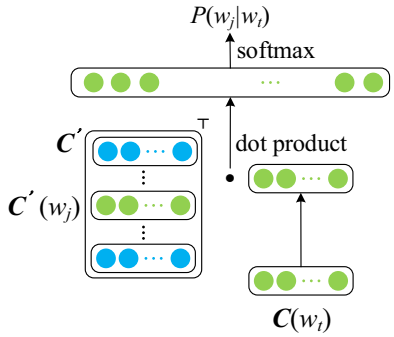

Fig. 4 Neural network architecture of Skip-gram model, where $\boldsymbol{C'}$ is the output embedding matrix containing vectors for context words. font size in figs differs

NNLM, except that the non-linear hidden layer is removed and the distributed embeddings of input words are averaged as the context representation. More formally, let $c_t = (w_{t-d}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+d})$ be the context of word w_t, $\boldsymbol{C}$ and $\boldsymbol{C'}$ be the input and output vector representation matrix respectively, where $\boldsymbol{C}$ is the embedding matrix to be learned. And let $\boldsymbol{C}(c_t)$ be the averaged vector of embeddings of the words in c_t which can be regarded as the context representation. Then the probability of word w_t given context c_t is:

$$P(w_t|c_t) = \frac{\exp(\boldsymbol{C'}(w_t)^\top \boldsymbol{C}(c_t))}{\sum_{i=1}^{|V|} \exp(\boldsymbol{C'}(w_i)^\top \boldsymbol{C}(c_t))},$$

where $|V|$ is the vocabulary size. And the objective of the model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T} \log P(w_t|c_t).$$

Correspondingly, as illustrated in Figure 4, the Skip-gram model predicts the context words given the central one. Therefore the probability of context word w_j given central word w_t is:

$$P(w_j|w_t) = \frac{\exp(\boldsymbol{C'}(w_j)^\top \boldsymbol{C}(w_t))}{\sum_{i=1}^{|V|} \exp(\boldsymbol{C'}(w_i)^\top \boldsymbol{C}(w_t))}.$$

And it maximizes the objective function

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j\in c_t} \log P(w_j|w_t).$$

Since the hidden layer of NNLM is removed in these two models, the number of parameters is significantly reduced. More specifically, they only need to update the two matrices $\boldsymbol{C}$ and $\boldsymbol{C'}$ while training. Therefore, the computational complexity is effectively reduced.

Later, several techniques have been used to further reduce the computational cost of these two models [76]. Firstly, *Hierarchical Softmax* was applied as an efficient approximation of the full softmax, where a binary Huffman tree is built to take the place of the softmax layer, with its leaves being words in the vocabulary and each inner node representing the relative probabilities of the child nodes.

Formally, each word w is reachable via an appropriate path from the root of the tree. Let $n(w, k)$ be the k-th node on the path to w, and $L(w)$ be the length of this path. Additionally, let $f(n(w, k))$ be 1 if $n(w, k+1)$ is the right child of $n(w, k)$ and -1 otherwise. Take the Skip-gram model for instance, the $P(w_j|w_t)$ is then defined as follows:

$$P(w_j|w_t) = \prod_{k=1}^{L(w_j)-1} \gamma(f(n(w_j,k)) \cdot \boldsymbol{C'}(n(w_j,k))^\top \boldsymbol{C}(w_t),$$

where $\gamma(x) = 1/(1+\exp(-x))$, and $\boldsymbol{C'}(n)$ is the representation vector of the inner node n. By applying hierarchical softmax, only $\log_2(|V|)$ nodes need to be evaluated instead of $|V|$ output nodes in the full softmax. Besides, since the Huffman tree assigns short codes to frequent words, the training process is further accelerated.

Secondly, *Negative Sampling* was proposed to reduce the number of negative samples and thus speed up the model. Specifically, instead of considering all negative samples, only k randomly selected negative samples are used, therefore, in the Skip-gram model, the probability becomes:

$$P(w_j|w_t) = \frac{\exp(\boldsymbol{C'}(w_j)^\top \boldsymbol{C}(w_t))}{\sum_{i=1}^{k+1} \exp(\boldsymbol{C'}(w_i)^\top \boldsymbol{C}(w_t))},$$

where $w_2, w_3, \ldots, w_{k+1}$ are randomly selected negative samples, and $w_1 = w_j$ is the positive sample

In addition, the *Subsampling* technique is used to counter the imbalance between rare and frequent words by

discarding words with probability computed with the word frequency.

In practice, Word2Vec [76] is the most popular implement for both CBOW and Skip-gram.

2.2.4 GloVe and fastText

The CBOW model and Skip-gram model are regarded as a milestone in the development of distributed representation. Following them comes a great amount of works trying to learn better distributed representations or so-called word embeddings. Here we only introduce two that have marked impact on the community, namely the *Global Vector* (GloVe) model [86] and the *fastText* model [9, 55].

In order to make full use of the statistics of word co-occurrence in a corpus, the GloVe model directly captures such global information. Formally, we denote the word co-occurrence count matrix by $\boldsymbol{X}$ and the number of times w_j appears in the context of w_i by $\boldsymbol{X}_{ij}$. Then the loss function is defined as follows:

$$L = \sum_{i,j=1}^{|V|} g(\boldsymbol{X}_{ij})(\boldsymbol{C}(w_i)^\top \boldsymbol{C}(w_j) + \boldsymbol{b}_i + \boldsymbol{b}_j - \log \boldsymbol{X}_{ij})^2,$$

where $|V|$ is the vocabulary size, and $g(x)$ is a weighting function which mitigates the imbalance of rare and frequent words.

Popular embedding models assign a distinct vector to each word, and thus ignore the morphology of words. To overcome this limitation, fastText introduces an extension of the continuous Skip-gram model by learning representations for character n-grams and representing words as the sum of the n-gram vectors. fastText also leverages training tricks, such as hierarchical softmax and hashing n-gram features, to significantly accelerate the training process.

2.3 Polysemy problem

Although the low-dimensional dense distributed representations introduced above have achieved great success in the NLP community, a challenging problem, namely the polysemy, still remained unsolved. This is because all the works introduced assume that a word is represented by a static word embedding, in other words, each word is represented by a single prototype vector that does not change with its context. Under this situation, for a polysemous word, it is impossible to distinguish the exact meaning given its context, which may possibly be regarded as the intrinsic drawback of such static word embeddings. Yet a number of works have made efforts to mitigate the polysemy problem with the static representation.

An intuitive way to solve this problem is using multiple prototypes instead of a single prototype to represent a word. Following this idea, Reisinger and Mooney [95] presented a multi-prototype vector-space model, where multiple sense-specific vectors are produced for each word by clustering. More specifically, they first cluster a word's contexts to generate groups of similar context vectors, then compute an average prototype vector for each cluster, resulting in multiple vectors for each word. Eventually, the context-dependent representation of a word is determined by selecting one of its vectors that minimizes the distance to the current context vector.

The idea of multi-prototype was then widely adopted to learn sense-level embeddings. Huang et al. [50] adopted it in the architecture of SENNA and yielded promising results. Concretely, they first represent each context with a weighted average of the word embeddings in it, and then cluster these context by their vector. Finally, each word in the corpus is relabeled by the corresponding cluster and used to train the new word embedding for that cluster.

A deficiency of the above cluster-based models is the determination of the cluster number, which was simply predefined and may introduce mistakes. To cope with this problem, Chen et al. [15] proposed a model for joint word sense representation and disambiguation that exploits knowledge bases (e.g., WordNet [78]) which help to assign vector representations for senses. They further modified the architecture of the Skip-gram model to predict the context words as well as their senses, and trained the model with the relevant occurrences obtained by word sense disambiguation to learn word vectors and sense vectors jointly.

Similarly, Neelakantan et al. [83] also modified the Skip-gram model and learned word sense representation and discrimination jointly. During the training procedure of a token, the average of its context words' vectors are used to select the corresponding sense on which a gradient update is then performed. Besides, they further proposed a non-parametric multi-sense Skip-gram model, which learns varying number of senses for each word.

At the same period, Tian et al. [108] introduced a novel probabilistic framework to the Skip-gram model and employed hidden variables to determine the prototype of each word in the context. They then designed an effective Expectation-Maximization (EM) algorithm to train the proposed multi-prototype Skip-gram model.

Motivated by the intuition that a polysemous word in one language may have more than one different translations in a foreign language, Guo et al. [41] exploited bilingual parallel data to train sense-specific embeddings in the RNNLM architecture.

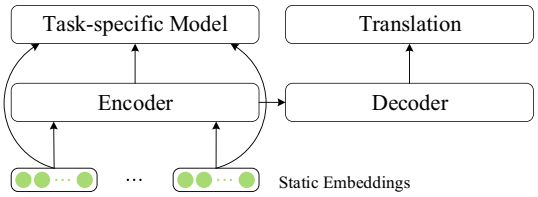

Fig. 5 Neural network architecture of CoVe

3 Dynamic representation

An alternative and possibly more effective way to resolve the polysemy problem is by using dynamic representations or so-called contextualized embeddings, where the representation changes with the context.

3.1 CoVe

In this line of work, McCann et al. [71] firstly proposed an inspiring *context vector* (CoVe) model where a deep long short-term memory (LSTM) encoder trained in a sequence-to-sequence machine translation (MT) model is used to generate contextualized embeddings which are then applied to a wide range of NLP tasks. From the architecture depicted in Fig. 5, we can say that this is a rather intuitive and straightforward model. More specifically, it directly takes the two-layer unidirectional LSTM encoder of an MT model to encode the pre-trained static GloVe embeddings and use the outputs as context vectors which are then concatenated with the GloVe ones and fed to downstream NLP tasks. CoVe has brought improvements in a wide variety of NLP tasks including sentiment analysis, question classification, entailment and question answering. More importantly, it has shed a light on the dynamic representation of words.

3.2 ELMo

A possible deficiency of CoVe is that it depends on cross-lingual data whose amount is limited. So there comes the *Embeddings from Language Models* (ELMo) [87], which brings back the task of language modeling to exploit the almost infinite unlabeled data and has captured the attention of the whole NLP community by the amazing success in a broad range of downstream tasks. ELMo representations are extracted from the internal layers of a deep bidirectional language model (biLM) pre-trained on a large unlabeled corpus.

Formally, let $(w_1, w_2, \ldots, w_N)$ be a sequence of N words. A forward language model computes the probability of a word w_t given previous words $(w_1, \ldots, w_{t-1})$, whereas the backward language model runs in the opposite direction,

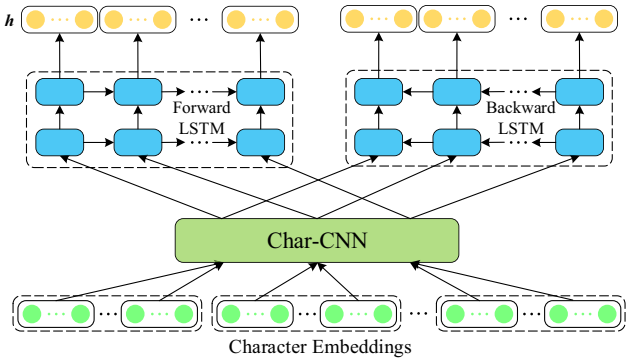

Fig. 6 Neural network architecture of ELMo. Char-CNN stands for character CNN

predicting the word w_t given its future context $(w_{t+1}, \ldots, w_N)$. The biLM jointly maximizes the log likelihood of the forward and backward ones:

$$L = \sum_{t=1}^{N} (\log P(w_t | w_1, w_2, \ldots, w_{t-1}) + \log(P(w_t | w_{t+1}, w_{t+2}, \ldots, w_N)).$$

It is clear from the architecture illustrated in Fig. 6 that ELMo employs a character convolutional neural network (CNN) to construct the word representations based on character embeddings, which not only successively mitigates the out-of-vocabulary (OOV) problem but also effectively reduces the amount of parameters. It further applies a two-layer bidirectional LSTM to capture the contextualized representations of the words, which are then used to compute the probability with a softmax layer.

The ELMo representation is a weighted combination of the intermediate layer representations in the biLM. Assume the biLM has L-layer LSTMs, then the neural network produces $2L + 1$ representations for word w_t:

$$R_t = \{\boldsymbol{x}_t, \overrightarrow{\boldsymbol{h}}_{t,j}; \overleftarrow{\boldsymbol{h}}_{t,j} \mid j = 1, \ldots, L\} = \{\boldsymbol{h}_{t,j} \mid j = 0, \ldots, L\},$$

where $\boldsymbol{h}_{t,0} = \boldsymbol{x}_t$ is the output of the character CNN, and $\boldsymbol{h}_{t,j} = [\overrightarrow{\boldsymbol{h}}_{t,j}; \overleftarrow{\boldsymbol{h}}_{t,j}]$ is the concatenation of the j-th-layer hidden representations of forward and backward LSTMs. When applying the ELMo to downstream tasks, the simplest way is to select the top layer $\boldsymbol{h}_{t,L}$. More generally, you can learn a task specific weighting of all the layers:

$$\boldsymbol{ELMo}_t = \lambda \sum_{j=0}^{L} s_j \boldsymbol{h}_{t,j},$$

where $\boldsymbol{s}$ are softmax-normalized weights and λ provides scalability to the whole ELMo vector.

ELMo effectively exploits the almost infinite unlabeled textual data to learn dynamic representations, and has significantly improved the state of the art in six NLP problems including question answering, textual entailment, semantic role labeling, named entity extraction, co-reference resolution and sentiment analysis. To demonstrate the capability of ELMo in resolving the polysemy problem, the authors first compared it to a powerful static word representation model (i.e. GloVe) by calculating the nearest neighbors of a highly polysemous word *play*, where ELMo showed a stronger ability to disambiguate word senses based on the context. It is further tested in SemCor 3.0, a word sense disambiguation (WSD) benchmark, where it achieved competitive performances with a state-of-the-art WSD-specific supervised model [52].[5] Additionally, it is also worth noting that with two intrinsic evaluations, the authors show that syntactic information is better captured by lower layers while semantic one by the higher layers.

The great success of ELMo has captured the imagination of the NLP community and inspired an amount of meaningful research concerning the contextualized representation of words. Most of these works focus on transferring linguistic information captured from large unlabeled text to downstream tasks. They achieve this by pretraining a language model on large unlabeled data and finetuning it on each specific task. Therefore, this line of works is also called *language model pretraining*, and in most of them the contextualized representations are regarded as by-products, just as same as what happened in the early age of static ones. Also there are attempts to learn representation using not only unlabeled text, but also labeled data. Clark et al. [17] propose a *Cross-View Training (CVT)* model to improve the representations of a Bi-LSTM sentence encoder in a semi-supervised manner.

3.3 ULMFit

At the same period, the *Universal Language Model Fine-tuning* (ULMFit) [49] method was proposed, which also employs LSTM-based language modeling to exploit large unlabeled data. Specifically, ULMFit consists of three steps:

a. A language model is trained on a large general-domain corpus to learn general linguistic features.
b. The language model is then fine-tuned on the target task corpus to capture task-specific information.
c. Eventually it is further fine-tuned with target classification task object.

Besides, two effective techniques, namely *discriminative finetuning* and *slanted triangular learning rates*, have been proposed to facilitate the target-domain language model fine-tuning process, which we describe in the following.

Intuitively, different layers of neural networks have different meanings (e.g., lower layers represent syntactic information while higher layers capture semantic one). Therefore, they should have separate learning rates, which is the key idea of discriminative finetuning. Formally, when training a model with parameters θ which are split into $\{\theta^1, \ldots, \theta^L\}$, the stochastic gradient descent (SGD) update at time t is as follows:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} L(\theta),$$

where $\nabla_{\theta^l} L(\theta)$ is the gradient with regard to the objective function L, η^l is the learning rate of the l-th layer.

Ideally, the model should maintain the information learned from the general-domain data while learning new features from target-domain. In addition, it should also converge quickly to an appropriate region at the beginning of training and steadily refine the parameters. To this end, the slanted triangular learning rates are proposed, where the learning rate first linearly increases and then linearly decays. The initial small learning rate helps to retain knowledge in the pretrained parameter.

Eventually, to cope with the catastrophic forgetting problem, the *gradual unfreezing* mechanism was proposed, where the pre-trained model is gradually unfrozen from the last layer.

With the three novel finetuning techniques which are then widely used in the following research, the ULMFit significantly outperforms the state-of-the-art on six text classification tasks.

3.4 GPT

Both ELMo and ULMFit have successfully learned linguistic information from large unlabeled data to help downstream tasks by pretraining language models and finetuning them on target tasks. However, their usage of the LSTM networks has limited the ability of their models to capture long-range dependencies. To deal with this problem, Radford et al. [90] employed the Transformer networks to learn a longer-range linguistic structure and proposed the *Generative Pre-Training* (GPT) model.

Figure 7 illustrates the architecture of the GPT model, where a multi-layer left-to-right Transformer [114] (or so-called *Transformer decoder*) is used instead of LSTM for the language model. Compared to LSTM, an obvious drawback of Transformer is that it does not model positional information. To cope with this, learned position embeddings are summed with the word embeddings to be the input to the Transformer, where first multi-head self-attention operations

[5] Please refer to the paper [87] for detailed results.

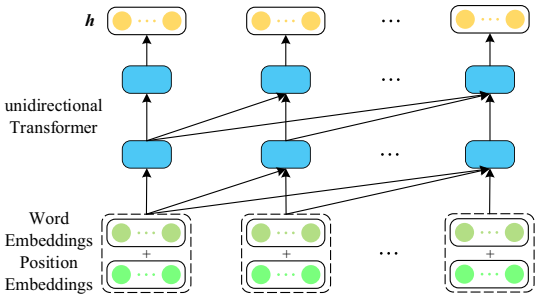

Fig. 7 Neural network architecture of GPT. The input word and position embeddings are summed

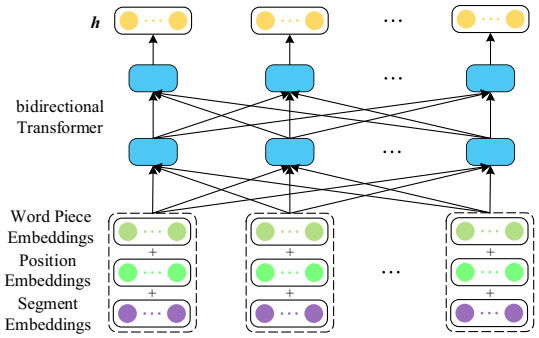

Fig. 8 Neural network architecture of BERT. The input word piece, position and segment embeddings are summed

and then feed-forward layers are applied to it to predict the probability distribution of the next word.

After pretraining, the model is further fine-tuned on the supervised target task with both language modeling and target task loss. Let $\mathcal{D}$ be a labeled dataset, where each instance consists of a word sequence $x = \{w_1, \ldots, w_n\}$ and a label y. Then the language modeling objective for $\mathcal{D}$ is:

$$L_1(\mathcal{D}) = \sum_x \sum_i \log P(w_i | w_{i-k}, \ldots, w_{i-1}),$$

where k is the context window size.

For the target task, the inputs x are fed to the pre-trained model to obtain the activation h_l^n, which is then passed through an extra feed-forward layer and a softmax layer to predict y:

$$P(y|w_1, \ldots, w_n) = \sigma(h_l^n W_y),$$

where σ is the softmax function. Then the objective for the target task is as follows:

$$L_2(\mathcal{D}) = \sum_{(x,y)} \log P(y|w_1, \ldots, w_n).$$

Differently from ULMFit where L_1 and L_2 are optimized separately, here they are optimized jointly with a weight λ:

$$L(\mathcal{D}) = L_2(\mathcal{D}) + \lambda L_1(\mathcal{D}).$$

Another significant contribution of GPT is that it has constructed a general framework based on the pre-trained language model that is capable of solving a wide range of NLP tasks including textual entailment, similarity, question answering and commonsense reasoning.

Specifically, they convert the structure of inputs from different tasks into ordered sequences by adding randomly initialized start, end and delimiter tokens so that the pre-trained model can process them. For instance, in textual entailment tasks, they concatenate the premise and hypothesis token sequences, with a delimiter token in between.[6]

With the powerful representation ability of Transformer and the general framework, GPT has amazingly achieved state of the art on nine out of 12 NLP tasks and has further justified the idea of language model pretraining and contextualized representation.

GPT 2.0 [91] is an upgraded version of GPT, that achieves state of the art results on seven out of eight tested language modeling datasets in a zero-shot setting. It contains more parameters (1.5B) and is trained on a larger dataset. Further, to empower the model with transferring and zero-shot ability, GPT 2.0 is trained with multi-task objects.

3.5 BERT

Despite the impressive success, GPT as well as other language model-based approaches are restricted by the unidirectional objective of language model, which limits the view of the model to left or right context only during pretraining. To fully develop the potential of the pre-trained contextualized representation, the *Bidirectional Encoder Representations from Transformers* (BERT) [26] was proposed, where both left and right contexts are captured by a deep bidirectional Transformer (or so-called *Transformer encoder*) to pre-train stronger contextualized representations as illustrated in Fig. 8.

Concretely, BERT is pre-trained with two novel unsupervised tasks, namely *Masked Language Model* (MLM) and *Next Sentence Prediction* (NSP), which we will introduce in follows.

The objective of MLM is to predict the tokens randomly masked given its context, which allows the model to capture both left and right contexts. Practically, the tokens are

[6] Please refer to the paper [90] for a detailed description of the preprocessing techniques.

masked in a quite complicated and heuristic way. 15% of them are masked at random. In 80% of the time, they are replaced with an artificial token `[MASK]`, in 10% of the time, they are replaced with a random token, while in the rest of the time, they are kept unchanged. It is worth noting that this objective is similar to that of the SENNA, where the left and right contexts are used to predict the central token.

In order to capture the relationship between sentences which is not directly modeled by the MLM, the NSP objective is proposed to predict whether a sentence *A* actually follows sentence *B*. Where the training data can be trivially generated from any monolingual corpus. This NSP objective is optimized jointly with the MLM one, and has been demonstrated highly beneficial to Question Answering (QA) and Natural Language Inference (NLI) tasks.

Another difference from GPT is that BERT uses WordPiece embeddings [122] instead of word ones, which are summed with the position and segment (denotes whether a token belongs to the first or second sentence) embeddings as the input.

BERT inherited and developed the idea of GPT to solve various types of NLP tasks in a general framework. Firstly, to deal with different input data structures, it used a sentence separator token (`[SEP]`) to separate sentences *A* and *B*, and a classifier token (`[CLS]`) at the beginning of each sequence as an aggregated representation for classification tasks.[7] Then the model for a particular target task is designed by incorporating BERT with a task-specific output layer. Four architectures are illustrated in the paper for four kinds of general NLP tasks. For instance, in sentence classification tasks, the final hidden state of the classifier token `[CLS]` is passed through a softmax layer to obtain the probability distribution. While in sequence labeling tasks, the final hidden state of each individual token is used for classification.

With stronger capability of capturing information from both sides, BERT has overwhelmingly beaten previous state-of-the-art on 11 NLP tasks and has inspired several interesting works trying to enhance it in different ways. The most explored direction is introducing different pretraining objectives. *BERT-wwm* [22] and *ERNIE (Baidu)* [105] strengthened generalization by predicting masked whole words/entities instead of word pieces. *SpanBert* [54] improves span prediction results by masking continues multi-word as a span and predicting entire content within a span. Instead of predicting masked tokens, *ELECTRA* [92] pretrain model with replaced token detection task. *Albert* [61] replaced the next sentence prediction with a novel sentence order prediction task. These harder pretraining tasks better exploit the potential of the deep transformer models. *MASS* [102] achieved significant improvements on generation tasks by reconstructing a sentence from sentence fragments. Another research line improves BERT performance by combining it with knowledge graphs. *ERNIE (THU)* [128] and *KnowBERT* [88] incorporated BERT with entity representations. *K-BERT* [65] injected triples of knowledge graph into the sentences as domain knowledge. Due to the large scale of BERT, there are also important researches attempted to reduce both the model size and computation cost of BERT. *DistillBERT* leveraged knowledge distillation to train a smaller but still effective BERT model. *Albert* reduced BERT parameter by sharing parameters across layers. Other BERT modifications for specific purposes, such cross-lingual and multi-modal, are included in Sects. 5.2 and 6.[8]

3.6 XLNet

Compared to the previous *autoregressive* (AR) methods which estimate the probability distribution of a text corpus with autoregressive models (e.g., ELMo, GPT), BERT is regarded as an *autoencoding* (AE) pretraining method, whose goal is to reconstruct the original data from corrupted input. Since the density estimation is excluded from the objective of BERT, it can readily utilize bidirectional contexts for reconstruction. And this may possibly explain its empirical superiority to previous AR methods. However, the artificial symbols like `[MASK]` introduced in pretraining are absent during finetuning, resulting in a pretrain-finetune discrepancy. Moreover, BERT assumes the predicted tokens are independent of each other, which may weaken its ability to model high-order, long-range dependencies that are prevalent in natural language.

To cope with such problems, Yang et al. [124] proposed XLNet, a generalized AE pretraining method based on Transformer-XL [24], which learns from bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and breaks BERT's limitations with its autoregressive formulation. XLNet employs the permutation language modeling objective, which brings advantages of both AR and AE methods while avoiding their weakness. Figure 9 illustrates an example of predicting the token w_3 with different factorization orders of a same input sequence in the permutation language model. Besides, to properly calculate the context representation required by the probability function, the authors proposed the Two-Stream Self-Attention to replace standard self-attention in the Transformer architecture,

[7] Such functional tokens are also used by GPT but only introduced while finetuning.

[8] Check https://github.com/thunlp/PLMpapers and https://github.com/cedrickchee/awesome-bert-nlp for latest progress of dynamic word representation.

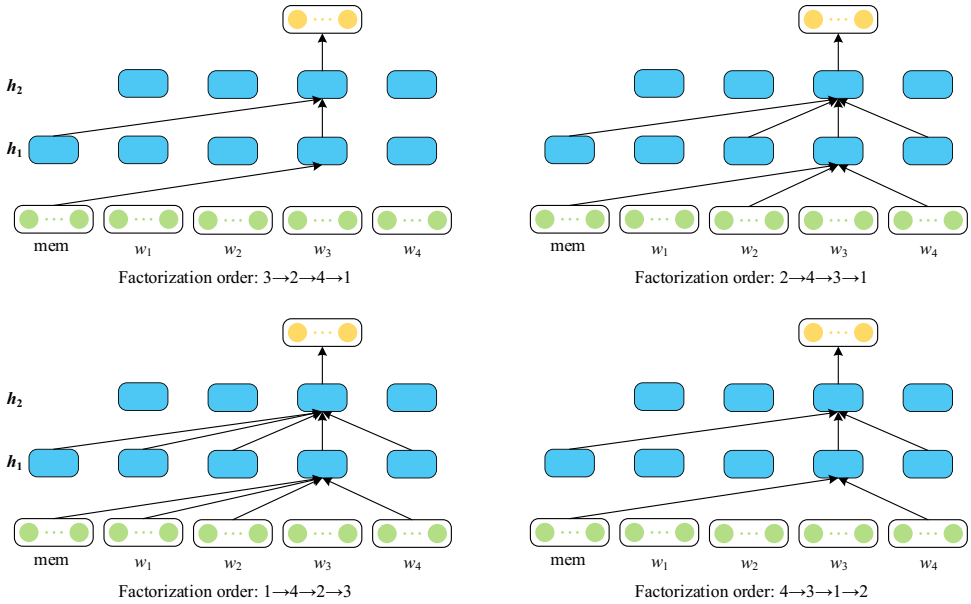

Fig. 9 Illustration of the permutation language modeling objective for predicting w_3 with different factorization orders

Table 2 Summary of word representation models, which are generally arranged in chronological order. Two sub-tables separated by a solid line show the static models and the dynamic models, respectively. *LM* denotes language mode here. *NSP* denotes next sentence prediction. *SOP* denotes sentence order prediction

Name	Objective	Model	Technique
One-hot	–	–	Represent words using high-dimensional, discrete, orthogonal vectors
Distributional	–	–	Use word co-occurrence statistics to model semantic distance
NNLM	LM	Feed-forward NN	Directly learn low-dimensional, dense and continuous vectors
SENNA	LM	Feed-forward NN	Pretrain embedding on unlabeled data to benefit downstream tasks
CBOW and Skip-gram	LM	Logistic regression	Accelerate computation by removing hidden layer
GloVe	LM	Logistic regression	Additionally leverage word co-occurrence information
fastText	LM	Logistic regression	Consider morphology by representing words with n-gram characters Leverage tricks to accelerate training
CoVe	Translation	LSTM	Propose contextual embedding using cross-lingual corpus
ELMo	LM	bi-LSTM	Consider bidirectional context
ULMFit	LM	LSTM	Propose techniques to improve fine-tuning
GPT	LM	Transformer	Use transformer to model long context
GPT 2.0	LM Multi-task	Transformer	Larger model and more training data Use multi-task objects to benefit zero-shot tasks
BERT	Masked LM NSP	Transformer	Consider bidirectional contexts and relation of sentence pairs
XLNet	Permuted LM Remove NSP	Transformer-XL	Use Transformer-XL to model long texts Use Permuted LM to remove pretrain-finetune discrepancy
RoBERTa	Masked LM Remove NSP	Transformer	Leverage training tricks to exploit potential of BERT
ELECTRA	Replaced LM Remove NSP	Transformer	Improve training efficiency with harder training objectives
Albert	Masked LM SOP	Transformer	Share layer parameter and decompose embedding to reduce model size

where the content and query representations are computed separately.[9]

By integrating the advantages of both AR and AE based pretraining methods and avoiding their weakness, XLNet has outperformed BERT on 20 tasks, and achieves state-of-the-art on 18 tasks including questing answering, natural

[9] Please refer to the paper [124] for implementation details.

language inference, sentiment analysis and document ranking. To show the correlation among the word representation models introduced above, we present a summary of them in (Table 2).

4 Evaluation methods and datasets

Existing evaluation metrics of word embeddings fall into two categories, namely *intrinsic evaluation* and *extrinsic evaluation*, both of which have been widely discussed by previous literature [3, 6, 36, 37, 97]. Therefore, in this section, we provide a concise introduction to the two kinds of evaluations based on the works we have discussed above and refer the readers to previous researches for detailed description and analysis.

4.1 Intrinsic evaluation

Intrinsic evaluations directly assess the syntactic or semantic relationship between words based on artificial assessments. Based on how the assessments are obtained, such methods can be further classified as *absolute intrinsic evaluation* and *comparative intrinsic evaluation*. In the first category, the human assessments are collected in advance and then used as a benchmark for word embedding methods. While in the comparative intrinsic evaluation, accessors evaluate the quality of word embeddings directly based on their results on a specific word relation task.

The absolute intrinsic evaluations are more commonly employed than the comparative ones since they do not require human involvement while testing. In the following, we briefly introduce some popular evaluation methods.

Word semantic similarity method is one of the most broadly used methods, which compares the distance between words in the embedding space to human heuristic judgments. The widely used test sets of this kind include SimVerb-3500 [35], WordSim-353 [32] and MEN [12].

Word analogy method has soon become popular since it was brought out along with the prominent CBOW and Skip-gram models [74]. Here, given three words a, b and c, the embeddings are used to predict the word d such that the relation $a : b$ is the same as $c : d$. For instance, let $a = France$, $b = Paris$ and $c = Germany$, then $d = Berlin$. Popular test sets of this kind include Google Analogy [74], Microsoft Research Syntactic Analogies Dataset [77], WordRep [34] and BATS [38].

Synonym detection method evaluates the ability of embeddings to select the most synonymous word to a given word among several candidates. For instance, given a target word *levied*, one should choose between *imposed* (correct), *believed*, *requested* and *correlated*. Datasets that can be used in this method include TOEFL [62] , ESL [112] and RDWP [53].

Concept categorization method evaluates the word embedding space by clustering. Given a set of nominal words, the goal of the task is to categorize them into different subsets (e.g., *dogs* and *cats* goes to the *mammal* class, *pineapples* and *bananas* goes to the *fruit* class). Useful datasets include BM [5], AP [1] and ESSLLI-2008 [4].

Selectional preference method evaluates the ability of word embeddings to identify typical verb-noun pairs where the noun is for the verb either as a subject or as an object (e.g., *people* is frequently used as the subject rather than the object of the verb *eat*). Broadly used test sets here include MSTNN [72], UP [85] and GDS [40].

4.2 Extrinsic evaluation

Extrinsic evaluations use word embeddings as input to downstream tasks and measure changes with the specific metric of these tasks. Since word embeddings are applicable to almost any NLP task, theoretically all such tasks can be viewed as extrinsic evaluations. A subset of such downstream tasks includes Language Modeling (LM) [7, 73], Chunking [20, 110], Part-of-Speech Tagging [20], Named Entity Recognition (NER) [20, 86], Sentiment Analysis (SA) [97, 109], Semantic Role Labeling (SRL) [20], Dependency Parsing [13], Machine Translation (MT) [16], Nature Language Inference (NLI) [10, 121] and Machine Reading Comprehension (MRC) [93, 94].

An implicit assumption lying under such extrinsic evaluations is that word embeddings bringing a good result on one task will perform well on others, which has been questioned by previous literature [97]. And it has been demonstrated empirically that different NLP tasks favor different word embeddings. Therefore, although the extrinsic evaluations are useful for comparing embeddings relatively on a specific task, they are not recommended as metrics for the general assessment of embedding quality.

5 Cross-lingual word embedding

There are about 7000 different languages in the world, but only a few languages have rich human annotated resources. This calls for cross-lingual transfer learning of word embeddings, where the model trained on rich-resource languages is applied to low-resource ones with the input embeddings projected to a shared semantic space. Such embeddings are therefore called *cross-lingual word embeddings*.

Based on the type of monolingual embeddings used, cross-lingual embedding learning methods can be categorized as static and dynamic ones, where the first has been

broadly explored by previous literature and the second has been recently investigated by several works. Besides, such approaches can also be classified into *on-line* and *off-line* ones according to the training objective. Typically, on-line approaches optimize monolingual and cross-lingual objectives jointly, while off-line approaches take pre-trained monolingual word embeddings of different languages as input and project them into a shared semantic space.

In this section, we provide a brief introduction of these methods and refer the readers to previous literature [96, 113] for detailed description and analysis.

5.1 Static cross-lingual word embedding

For static embeddings, the on-line methods typically learn language models for the source and target languages and jointly optimize their objectives with a cross-lingual objective. For instance, Klementiev et al. [56] and Kočiský et al. [57] proposed to learn bilingual embeddings and word alignments jointly. Later, Luong et al. [69] extended the monolingual skip-gram model to learn bilingual embeddings along with both sentence and word-level alignments. Hermann and Blunsom [45] presented a model that leverages the semantic similarity of aligned sentence pairs to learn bilingual embeddings. Vulić and Moens [116] proposed an approach to exploit document-aligned comparable corpora to learn bilingual embeddings by first merging two aligned documents into a pseudo-bilingual document and then training a skip-gram model on it.

Off-line methods aim at learning a projection (mostly linear transformation matrix) to transform the vector space of a source language to the vector space of the target language. Such a matrix can be learned in a supervised way by minimizing the squared Euclidean distance (mean squared error, MSE) between the transformed source embedding of word w and the target embedding of its translation [75]. The matrix can also be obtained by optimizing a max-margin hinge loss instead of MSE [63] or with singular value decomposition (SVD) [100, 123].

Faruqui and Dyer [29] proposed another way to project embeddings from both source and target languages to a shared space with canonical correlation analysis (CCA). Guo et al. [42–44] explored cross-lingual embedding for dependency parsing with both full lexical features and additional useful features such as cross-lingual word clusters. Lu et al. [67] extended their work by introducing nonlinearity in the mapping process.

Recently, a line of unsupervised cross-lingual embedding learning methods has also yielded promising results. Lample et al. [60] proposed to first build a bilingual dictionary with adversarial learning and then obtain bilingual embeddings with a refinement method. Following this work, Chen and Cardie [14] proposed a similar two-step framework to learn multilingual embeddings while leveraging the interdependencies existing among many languages that previous work ignored. In order to mitigate the instability issues in learning cross-lingual embeddings for distant language pairs, Heyman et al. [47] proposed a robust framework that learns a shared multilingual embedding space by incrementally adding new languages to the current space.

5.2 Dynamic cross-lingual word embedding

Inspired by the great potential that monolingual dynamic word embeddings have shown, several works have recently been proposed to connect the rich expressiveness of dynamic word embeddings with cross-lingual transfer.

In the category of on-line methods, Mulcaire et al. [81] based their model on ELMo [87] and proposed a polyglot contextual representation model by capturing character-level information from multilingual data. Lample and Conneau [58] adapted the objectives of BERT [26] and exploited cross-lingual supervision from parallel data to learn cross-lingual language models (XLMs), which have obtained state-of-the-art results on several cross-lingual tasks. Later, they further showed that multilingual language models pre-trained at scale lead to significant performance gains in a great number of cross-lingual transfer tasks, showing the possibility of multilingual modeling without sacrificing per-language performance [21].

Whereas for off-line methods, Mulcaire et al. [98] aligned pre-trained contextualized word embeddings through linear projection in an off-line fashion. They used the averaged contextualized embeddings as an anchor for each word, and learn the transformation matrix in the anchor space. Wang et al. [120] proposed to learn this transformation directly in the contextual space, and hence produced word sense-preserving cross-lingual dynamic embeddings. Mulcaire et al. [82] assessed recent approaches to dynamic cross-lingual embeddings and showed that they can greatly facilitate cross-lingual dependency parsing. They also demonstrated that on-line methods better encode cross-lingual lexical correspondence compared to off-line ones.

5.3 Multilingual embedding

In addition to transferring rich resource language embedding to low-resource language with projection, there are also attempts to simultaneously train embedding in different languages. Devlin et al. [26] pre-trained a single language model (Multi-BERT) on the concatenation of monolingual Wikipedia corpora from 104 languages, which is surprisingly good at zero-shot cross-lingual model transfer. Pires et al. [89] showed with a broad range of probing experiments that Multi-BERT is able to transfer between languages

even with zero lexical overlap by successfully capturing multilingual representations. Artetxe et al. [2] investigated the generalization ability of Multi-BERT by designing an alternative approach that transfers a monolingual model to new languages at the lexical level. The results contradict the common beliefs that Multi-BERT generalizes well because of the use of shared subword vocabulary and joint training across multiple languages. Instead, they suggest that the monolingual models learn abstractions which generalize across languages.

6 Discussions and open issues

Although dynamic word representations have achieved such amazing success in a majority of NLP tasks, a great number of problems in this field still remain unexplored. In this section, we will introduce some of such promising yet challenging problems to indicate directions for the readers interested in them.

6.1 Self-supervised learning

Most of the prevailing pretraining methods (e.g. ELMo [87], GPT [90] and BERT [26]) are trained with self-supervision, where unlabeled corpora are reconstructed to obtain labeled data for some manually designed tasks. These tasks are often based on co-occurrence of words and sentences, while other lexical, syntactic and semantic information which might be more valuable has been ignored.

To this end, Sun et al. [106] has proposed a continual pretraining framework to learn such various tasks incrementally through constant multi-task learning (MTL). They have investigated several pretraining tasks with respect to word-level, structure-level and semantic-level and yielded promising results.

From another perspective, the previous tasks have been too easy for the pretraining models to learn better representations. Therefore, it is promising and challenging to design harder and more sophisticated tasks to enforce the model to learn stronger representations

6.2 Multi-task learning

Multi-task learning has been an active research concern in natural language processing for a long time. Due to its capability of leveraging large amounts of supervised data from many related tasks and alleviating over-fitting to a specific task, it has been increasingly employed to learn representations with neural networks.

Recently, Liu et al. [66] proposed to incorporate the prominent pretrained language model (i.e. BERT) into a multi-task deep neural network (MT-DNN) framework and obtained new state-of-the-art on ten natural language understanding tasks. While Clark et al. [19] introduced knowledge distillation, where single-task models teach a multi-task model, to finetune BERT on multiple tasks of the GLUE benchmark to enhance representation learning.

Both works have demonstrated empirically that MTL and language model pretraining are complementary to each other. Therefore, it is worth to conduct further research on better approaches of combining them.

6.3 Multi-modal learning

There is another line of works trying to extend the powerful transformer based pretrained language models, especially BERT, to enhance the learning of multi-modal representations.

For instance, Lu et al. [68] and Su et al. [103] proposed to learn joint representations of image and natural language by pretraining models modified from BERT on Conceptual Captions dataset [99] with three tasks, namely masked language modeling, masked visual-feature classification and sentence-image alignment. Whereas Sun et al. [104] adapted BERT to video domain and proposed to learn video and language representations jointly, yielding promising results in a variety of interesting text-video-related tasks.

Given the progress these works have made, more researches could be expected to jointly learn the representations of multiple modalities.

6.4 Few-shot learning

Human can learn from a limited number of examples, but most machine learning algorithm only succeeds with rich annotated data. Few-shot learning is proposed to reduce such a gap between humans and machines. Traditional methods depend highly on hand-crafted features [30, 31]. Recent efforts primarily focus on metric learning [48, 101, 115], which classify an item according to its distance with labeled examples. These metric based methods essentially learn a sample representation, which is word embedding for token level NLP tasks. This led to a huge need to design embedding dedicated to Few-shot learning cases. For example, Hou et al. [48] leveraged the special query-support structure in few-shot learning and proposed a pair-wise embedding to reduce representation ambiguity.

6.5 Language generation

The pretrained language models have also been employed in language generation tasks. For instance, Golovanov et al. [39] and Zhang et al. [126] have proposed to generate output sequences in an encoder-decoder framework based on GPT and BERT respectively. While Wang and Cho [118] regarded BERT as a Markov random field language model and generate sentences from it by Gibbs sampling. Zhang et al. [127] proposed a BERT-based automatic evaluation metric for text generation, which computes the sentence similarity as a weighted aggregation of cosine similarities between tokens.

6.6 Interpretability

As same as most of the neural models, these popular language model pretraining methods have also been criticized for their poor interpretability. Therefore, it is interesting and challenging to provide explanations for their incredible success.

In the line of such researches, Clark et al. [18] analyzed the attention mechanisms in BERT and suggested that the attentions are syntax-aware, which may have contributed to BERT's success. Hewitt and Manning [46] evaluated ELMo and BERT with a novel structural probe and showed that both of them have the entire syntax trees embedded implicitly in their representation space. Liu et al. [64] conducted a thorough analysis concerning the linguistic knowledge and transferability of the pretrained language models (i.e. ELMo, GPT and BERT), and drawn some interesting conclusions. Tenney et al. [107] quantified where specific types of linguistic information are encoded in BERT, and provided evidence corroborating that it can represent syntactic and semantic information while modeling complex interactions between different levels.

6.7 Adversarial attack

Although the language model pretraining methods have achieved incredible success in a great number of NLP tasks, what do the improvements truly come from has been questioned recently.

Niven and Kao [84] have shown that the BERT's peak performance on the Argument Reasoning Comprehension Task is achieved by exploiting the spurious statistical cues in the dataset. They further constructed an adversarial dataset which provides a more robust assessment of argument comprehension. And Wallace et al. [117] proposed a gradient-guided search to find input-agnostic sequences that trigger a model to produce some specific prediction when concatenated with any input from a dataset.

These adversarial attack methods help to probe the 'weakness' of datasets and make them harder to solve, which will eventually enforce the representation learning models become stronger.

6.8 Simplifying the traning process

Most of the existing language model pretraining methods require massive amounts of computing resources inaccessible to all but few. For instance, ELMo is trained on four NVIDIA P100 GPUs for 7 days, while BERT is trained on four cloud TPUs for 4 days. Therefore, it is challenging yet beneficial to simplify the models and thus reduce the compute and shorten the time required while training.

Recently, Yijia [125] proposed to model local context instead of global one to speed-up the training procedure and yielded competitive performances to ELMo.

7 Conclusion

This survey adopts a novel perspective, i.e. developing trend of static-to-dynamic, to present a comprehensive overview of word representation models. It has also covered the evaluation methods of these representations and their extension in cross-lingual scenario. Finally, we discussed the open issues and possible future work for word representation.

Funding This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011 and 61772153.

References

1. Almuhareb A (2006) Attributes in lexical acquisition. PhD thesis, University of Essex
2. Artetxe M, Ruder S, Yogatama D (2019) On the cross-lingual transferability of monolingual representations. arXiv preprint arXiv:1910.11856
3. Bakarov A (2018) A survey of word embeddings evaluation methods. arXiv preprint arXiv:1801.09536
4. Baroni M, Evert S, Lenci A (2008) Bridging the gap between semantic theory and computational simulations. In: Proc. of the esslli workshop on distributional lexical semantic. FOLLI, Hamburg
5. Baroni M, Murphy B, Barbu E, Poesio M (2010) Strudel: a corpus-based semantic model based on properties and types. Cogn Sci 34:222–254
6. Baroni M, Dinu G, Kruszewski G (2014) Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Baltimore, Maryland, pp 238–247. https://doi.org/10.3115/v1/P14-1023. https://www.aclweb.org/anthology/P14-1023

7. Bengio Y, Ducharme R, Vincent P, Janvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155
8. Blei DM, Ng AY, Jordan MI, Lafferty J (2003) Latent dirichlet allocation. J Mach Learn Res 3:2003
9. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
10. Bowman SR, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326
11. Brown PF, deSouza PV, Mercer RL, Pietra VJD, Lai JC (1992) Class-based n-gram models of natural language. Comput Linguist 18(4):467–479
12. Bruni E, Tran NK, Baroni M (2014) Multimodal distributional semantics. J Artif Int Res pp 1–47
13. Chen D, Manning C (2014) A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 740–750 https://doi.org/10.3115/v1/D14-1082. https://www.aclweb.org/anthology/D14-1082
14. Chen X, Cardie C (2018) Unsupervised multilingual word embeddings. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Brussels, Belgium, pp 261–270. https://doi.org/10.18653/v1/D18-1024. https://www.aclweb.org/anthology/D18-1024
15. Chen X, Liu Z, Sun M (2014) A unified model for word sense representation and disambiguation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1025–1035. https://doi.org/10.3115/v1/D14-1110. https://www.aclweb.org/anthology/D14-1110
16. Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. In: Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation. Association for Computational Linguistics, Doha, Qatar, pp 103–111.https://doi.org/10.3115/v1/W14-4012. https://www.aclweb.org/anthology/W14-4012
17. Clark K, Luong MT, Manning CD, Le QV (2018) Semi-supervised sequence modeling with cross-view training. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Brussels, Belgium, pp 1914–1925. https://doi.org/10.18653/v1/D18-1217. https://www.aclweb.org/anthology/D18-1217
18. Clark K, Khandelwal U, Levy O, Manning CD (2019a) What does BERT look at? An analysis of BERT's attention. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: analyzing and interpreting neural networks for NLP. Association for Computational Linguistics, Florence, Italy, pp 276–286. https://doi.org/10.18653/v1/W19-4828. https://www.aclweb.org/anthology/W19-4828
19. Clark K, Luong MT, Khandelwal U, Manning CD, Le QV (2019b) BAM! born-again multi-task networks for natural language understanding. In: Proc. of ACL, pp 5931–5937
20. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
21. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2019) Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116
22. Cui Y, Che W, Liu T, Qin B, Yang Z, Wang S, Hu G (2019) Pre-training with whole word masking for chinese bert. arXiv preprint arXiv:1906.08101
23. Dagan I, Pereira F, Lee L (1994) Similarity-based estimation of word cooccurrence probabilities. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Las Cruces, New Mexico, USA, pp 272–278. https://doi.org/10.3115/981732.981770
24. Dai Z, Yang Z, Yang Y, Cohen WW, Carbonell J, Le QV, Salakhutdinov R (2019) Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860
25. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407
26. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
27. Dinu G, Lapata M (2010) Measuring distributional similarity in context. In: Proc. of EMNLP, pp 1162–1172
28. Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1(3):211–218
29. Faruqui M, Dyer C (2014) Improving vector space word representations using multilingual correlation. In: Proceedings of the 14th conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Gothenburg, Sweden, pp 462–471. https://doi.org/10.3115/v1/E14-1049. https://www.aclweb.org/anthology/E14-1049
30. Fei-Fei L (2006) Knowledge transfer in learning to recognize visual objects classes. In: International Conference on Development and Learning. Department of Psychological and Brain Sciences, Indiana University, pp 1–8
31. Fink M (2005) Object classification from a single example utilizing class relevance metrics. In: Saul LK, Weiss Y, Bottou L (eds) Advances in neural information processing systems. MIT Press, pp 449–456. http://papers.nips.cc/paper/2576-object-classification-from-a-single-example-utilizing-class-relevance-metrics.pdf
32. Finkelstein L, Gabrilovich E, Matias Y, Rivlin E, Solan Z, Wolfman G, Ruppin E (2001) Placing search in context: the concept revisited. ACM Trans Inf Syst. https://doi.org/10.1145/503104.503110
33. Firth JR (1957) A synopsis of linguistic theory 1930–1955. In: Studies in linguistic analysis (special volume of the Philological Society), vol 1952–1959. The Philological Society, Oxford, pp 1–32. https://www.bibsonomy.org/bibtex/25e3d6c72cdd123a638f71886d78f3c1e/brightbyte
34. Gao B, Bian J, Liu TY (2014) Wordrep: A benchmark for research on learning word representations. arXiv preprint arXiv:1407.1640
35. Gerz D, Vulić I, Hill F, Reichart R, Korhonen A (2016) SimVerb-3500: A large-scale evaluation set of verb similarity. In: Proceedings of the 2016 conference on empirical methods in natural language processing. Association for Computational Linguistics, Austin, Texas, pp 2173–2182. https://doi.org/10.18653/v1/D16-1235. https://www.aclweb.org/anthology/D16-1235
36. Ghannay S, Favre B, Estève Y, Camelin N (2016) Word embedding evaluation and combination. In: Proceedings of the tenth international conference on language resources and evaluation (LREC'16). Portorož, Slovenia, pp 300–305 https://www.aclweb.org/anthology/L16-1046
37. Gladkova A, Drozd A (2016) Intrinsic evaluations of word embeddings: what can we do better? In: Proceedings of the 1st workshop on evaluating vector-space representations for NLP. Association for Computational Linguistics, Berlin, Germany, pp 36–42. https://doi.org/10.18653/v1/W16-2507. https://www.aclweb.org/anthology/W16-2507

38. Gladkova A, Drozd A, Matsuoka S (2016) Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In: Proceedings of the NAACL student research workshop. Association for Computational Linguistics, San Diego, California, pp 8–15. https://doi.org/10.18653/v1/N16-2002. https://www.aclweb.org/anthology/N16-2002
39. Golovanov S, Kurbanov R, Nikolenko S, Truskovskyi K, Tselousov A, Wolf T (2019) Large-scale transfer learning for natural language generation. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 6053–6058. https://doi.org/10.18653/v1/P19-1608. https://www.aclweb.org/anthology/P19-1608
40. Greenberg C, Demberg V, Sayeed A (2015) Verb polysemy and frequency effects in thematic fit modeling. In: Proceedings of the 6th workshop on cognitive modeling and computational linguistics. Association for Computational Linguistics, Denver, Colorado, pp 48–57. https://doi.org/10.3115/v1/W15-1106. https://www.aclweb.org/anthology/W15-1106
41. Guo J, Che W, Wang H, Liu T (2014) Learning sense-specific word embeddings by exploiting bilingual resources. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp 497–507. https://www.aclweb.org/anthology/C14-1048
42. Guo J, Che W, Yarowsky D, Wang H, Liu T (2015) Cross-lingual dependency parsing based on distributed representations. In: Proc. of ACL and IJCNLP, pp 1234–1244
43. Guo J, Che W, Yarowsky D, Wang H, Liu T (2016a) A distributed representation-based framework for cross-lingual transfer parsing. J Artif Int Res 55(1):995–1023
44. Guo J, Che W, Yarowsky D, Wang H, Liu T (2016b) A representation learning framework for multi-source transfer parsing. In: Proceedings of the thirtieth AAAI conference on artificial intelligence, AAAI'16. AAAI Press, Phoenix, Arizona, pp 2734–2740.
45. Hermann KM, Blunsom P (2014) Multilingual models for compositional distributed semantics. In: Proc. of ACL, pp 58–68
46. Hewitt J, Manning CD (2019) A structural probe for finding syntax in word representations. In: Proc. of NAACL, pp 4129–4138, https://doi.org/10.18653/v1/N19-1419
47. Heyman G, Verreet B, Vulić I, Moens MF (2019) Learning unsupervised multilingual word embeddings with incremental multilingual hubs. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 1890–1902. https://doi.org/10.18653/v1/N19-1188. https://www.aclweb.org/anthology/N19-1188
48. Hou Y, Zhou Z, Liu Y, Wang N, Che W, Liu H, Liu T (2019) Few-shot sequence labeling with label dependency transfer. arXiv preprint arXiv:1906.08711
49. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (vol 1: long papers), Association for Computational Linguistics, Melbourne, Australia, pp 328–339. https://doi.org/10.18653/v1/P18-1031. https://www.aclweb.org/anthology/P18-1031
50. Huang E, Socher R, Manning C, Ng A (2012) Improving word representations via global context and multiple word prototypes. In: Proceedings of the 56th annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 873–882
51. Huang F, Yates A (2009) Distributional representations for handling sparsity in supervised sequence-labeling. In: Proc. of ACL and IJCNLP, pp 495–503
52. Iacobacci I, Pilehvar MT, Navigli R (2016) Embeddings for word sense disambiguation: an evaluation study. In: Proc. of ACL, pp 897–907
53. Jarmasz M, Szpakowicz S (2003) Roget's thesaurus and semantic similarity. In: Proc. of RANLP, pp 212–219
54. Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O (2019) Spanbert: improving pre-training by representing and predicting spans. arXiv preprint arXiv:1907.10529
55. Joulin A, Grave E, Bojanowski P, Mikolov T (2017) Bag of tricks for efficient text classification. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics, vol 2, short papers. Association for Computational Linguistics, Valencia, Spain, pp 427–431. https://www.aclweb.org/anthology/E17-2068
56. Klementiev A, Titov I, Bhattarai B (2012) Inducing crosslingual distributed representations of words. In: Proceedings of COLING 2012. The COLING 2012 Organizing Committee, Mumbai, India, pp 1459–1474. https://www.aclweb.org/anthology/C12-1089
57. Kočiský T, Hermann KM, Blunsom P (2014) Learning bilingual word representations by marginalizing alignments. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, vol 2 short papers. Association for Computational Linguistics, Baltimore, Maryland, pp 224–229. https://doi.org/10.3115/v1/P14-2037. https://www.aclweb.org/anthology/P14-2037
58. Lample G, Conneau A (2019) Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291
59. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: Proc. of NAACL, pp 260–270
60. Lample G, Conneau A, Ranzato M, Denoyer L, Jégou H (2018) Word translation without parallel data. In: Proc. of ICLR
61. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942
62. Landauer TK, Dutnais ST (1997) A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol Rev 211–240
63. Lazaridou A, Dinu G, Baroni M (2015) Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In: Proc. of ACL and IJCNLP, pp 270–280
64. Liu NF, Gardner M, Belinkov Y, Peters ME, Smith NA (2019a) Linguistic knowledge and transferability of contextual representations. In: Proc. of NAACL, pp 1073–1094, https://doi.org/10.18653/v1/N19-1112
65. Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, Wang P (2019b) K-bert: Enabling language representation with knowledge graph. arXiv preprint arXiv:1909.07606
66. Liu X, He P, Chen W, Gao J (2019c) Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 4487–4496. https://doi.org/10.18653/v1/P19-1441. https://www.aclweb.org/anthology/P19-1441
67. Lu A, Wang W, Bansal M, Gimpel K, Livescu K (2015) Deep multilingual correlation for improved word embeddings. In: Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Denver, Colorado, pp 250–256. https://doi.org/10.3115/v1/N15-1028. https://www.aclweb.org/anthology/N15-1028

68. Lu J, Batra D, Parikh D, Lee S (2019) Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems. Curran Associates, Inc., pp 13–23. http://papers.nips.cc/paper/8297-vilbert-pretraining-task-agnostic-visiolinguistic-representations-for-vision-and-language-tasks.pdf
69. Luong T, Pham H, Manning CD (2015) Bilingual word representations with monolingual quality in mind. In: Proceedings of the 1st workshop on vector space modeling for natural language processing. Association for Computational Linguistics, Denver, Colorado, pp 151–159. https://doi.org/10.3115/v1/W15-1521. https://www.aclweb.org/anthology/W15-1521
70. McCallum A, Freitag D, Pereira FCN (2000) Maximum entropy markov models for information extraction and segmentation. In: Proceedings of the seventeenth international conference on machine learning, ICML '00. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 591–598
71. McCann B, Bradbury J, Xiong C, Socher R (2017) Learned in translation: contextualized word vectors. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural Information processing systems 30. Curran Associates, Inc., pp 6294–6305. http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf
72. McRae K, Ferretti TR, Amyote L (1997) Thematic roles as verb-specific concepts. Lang Cogn Process 12(2–3):137–176
73. Mikolov T, Karafiát M, Burget L, Cernocky J, Khudanpur S (2010) Recurrent neural network based language model. In: Kobayashi T, Hirose K, Nakamura S (eds) INTERSPEECH. ISCA, pp 1045–1048. https://www.bibsonomy.org/bibtex/2aee1e280d06e82474b17c4996aaea076/dblp
74. Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. ICLR Workshop
75. Mikolov T, Le QV, Sutskever I (2013b) Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168
76. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013c) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
77. Mikolov T, Yih Wt, Zweig G (2013d) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Atlanta, Georgia, pp 746–751. https://www.aclweb.org/anthology/N13-1090
78. Miller GA (1995) Wordnet: A lexical database for english. Commun ACM 39–41
79. Mnih A, Hinton G (2007) Three new graphical models for statistical language modelling. In: Proceedings of the 24th international conference on machine learning, ICML '07. Association for Computing Machinery, Corvalis, Oregon, USA, pp 641–648. https://doi.org/10.1145/1273496.1273577
80. Mnih A, Hinton GE (2009) A scalable hierarchical distributed language model. Adv Neural Inf Process Syst 21:1081–1088
81. Mulcaire P, Kasai J, Smith N (2019a) Polyglot contextual representations improve crosslingual transfer. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 3912–3918. https://doi.org/10.18653/v1/N19-1392. https://www.aclweb.org/anthology/N19-1392
82. Mulcaire P, Kasai J, Smith NA (2019b) Low-resource parsing with crosslingual contextualized representations. In: Proceedings of the 23rd conference on computational natural language learning (CoNLL), Association for Computational Linguistics, Hong Kong, China, pp 304–315. https://doi.org/10.18653/v1/K19-1029. https://www.aclweb.org/anthology/K19-1029
83. Neelakantan A, Shankar J, Passos A, McCallum A (2014) Efficient non-parametric estimation of multiple embeddings per word in vector space. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1059–1069. https://doi.org/10.3115/v1/D14-1113. https://www.aclweb.org/anthology/D14-1113
84. Niven T, Kao H (2019) Probing neural network comprehension of natural language arguments. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 4658–4664. https://doi.org/10.18653/v1/P19-1459. https://www.aclweb.org/anthology/P19-1459
85. Padó S, Lapata M (2007) Dependency-based construction of semantic space models. Comput Linguist 33(2):161–199
86. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1532–1543. https://doi.org/10.3115/v1/D14-1162. https://www.aclweb.org/anthology/D14-1162
87. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202. https://www.aclweb.org/anthology/N18-1202
88. Peters ME, Neumann M, Logan R, Schwartz R, Joshi V, Singh S, Smith NA (2019) Knowledge enhanced contextual word representations. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp 43–54. https://doi.org/10.18653/v1/D19-1005https://www.aclweb.org/anthology/D19-1005
89. Pires T, Schlinger E, Garrette D (2019) How multilingual is multilingual bert? arXiv preprint arXiv:1906.01502
90. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. https://s3-us-west-2amazonaws.com/openai-assets/research-covers/languageunsupervised/language/understanding/paper/pdf
91. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8)
92. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2019) Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683
93. Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) Squad: 100,000+ questions for machine comprehension of text. In: Proc. of EMNLP, pp 2383–2392
94. Rajpurkar P, Jia R, Liang P (2018) Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822
95. Reisinger J, Mooney RJ (2010) Multi-prototype vector-space models of word meaning. In: Proc. of HLT-NAACL, pp 109–117
96. Ruder S, Vulic I, Søgaard A (2017) A survey of cross-lingual embedding models. arXiv preprint arXiv:1706.04902
97. Schnabel T, Labutov I, Mimno D, Joachims T (2015) Evaluation methods for unsupervised word embeddings. In: Proceedings of

the 2015 conference on empirical methods in natural language processing. Association for Computational Linguistics, Lisbon, Portugal, pp 298–307. https://doi.org/10.18653/v1/D15-1036. https://www.aclweb.org/anthology/D15-1036
98. Schuster T, Ram O, Barzilay R, Globerson A (2019) Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 1599–1613. https://doi.org/10.18653/v1/N19-1162. https://www.aclweb.org/anthology/N19-1162
99. Sharma P, Ding N, Goodman S, Soricut R (2018) Conceptual captions: a cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proc. of ACL, pp 2556–2565, https://doi.org/10.18653/v1/P18-1238
100. Smith SL, Turban DHP, Hamblin S, Hammerla NY (2017) Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In: Proc. of ICLR
101. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp 4077–4087
102. Song K, Tan X, Qin T, Lu J, Liu T (2019) MASS: masked sequence to sequence pre-training for language generation. In: Proc. of ICML, pp 5926–5936
103. Su W, Zhu X, Cao Y, Li B, Lu L, Wei F, Dai J (2019) Vl-bert: Pre-training of generic visual-linguistic representations. arXiv preprint arXiv:1908.08530
104. Sun C, Myers A, Vondrick C, Murphy K, Schmid C (2019a) Videobert: A joint model for video and language representation learning. arXiv preprint arXiv:1904.01766
105. Sun Y, Wang S, Li Y, Feng S, Chen X, Zhang H, Tian X, Zhu D, Tian H, Wu H (2019b) Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223
106. Sun Y, Wang S, Li Y, Feng S, Tian H, Wu H, Wang H (2019c) Ernie 2.0: a continual pre-training framework for language understanding. arXiv preprint arXiv:1907.12412
107. Tenney I, Das D, Pavlick E (2019) BERT rediscovers the classical NLP pipeline. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 4593–4601. https://doi.org/10.18653/v1/P19-1452
108. Tian F, Dai H, Bian J, Gao B, Zhang R, Chen E, Liu TY (2014) A probabilistic model for learning multi-prototype word embeddings. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp 151–160. https://www.aclweb.org/anthology/C14-1016
109. Tsvetkov Y, Faruqui M, Ling W, Lample G, Dyer C (2015) Evaluation of word vector representations by subspace alignment. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Association for Computational Linguistics, Lisbon, Portugal, pp 2049–2054. https://doi.org/10.18653/v1/D15-1243. https://www.aclweb.org/anthology/D15-1243
110. Turian J, Ratinov LA, Bengio Y (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, Uppsala, Sweden, pp 384–394. https://www.aclweb.org/anthology/P10-1040
111. Turney PD (2001a) Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: De Raedt L, Flach P (eds) Machine learning: ECML 2001. Springer, Berlin Heidelberg, pp 491–502
112. Turney PD (2001b) Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: De Raedt L, Flach P (eds) Machine learning: ECML 2001. Springer, Berlin Heidelberg, pp 491–502
113. Upadhyay S, Faruqui M, Dyer C, Roth D (2016) Cross-lingual models of word embeddings: an empirical comparison. In: Proceedings of the 54th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Berlin, Germany, pp 1661–1670. https://doi.org/10.18653/v1/P16-1157. https://www.aclweb.org/anthology/P16-1157
114. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems. Curran Associates, Inc., pp 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
115. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D (2016) Matching networks for one shot learning. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in neural information processing systems. Curran Associates, Inc., pp 3630–3638. http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf
116. Vulić I, Moens MF (2015) Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, vol 2 (short papers). Association for Computational Linguistics, Beijing, China, pp 719–725. https://doi.org/10.3115/v1/P15-2118. https://www.aclweb.org/anthology/P15-2118
117. Wallace E, Feng S, Kandpal N, Gardner M, Singh S (2019) Universal adversarial triggers for attacking and analyzing NLP. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp 2153–2162. https://doi.org/10.18653/v1/D19-1221. https://www.aclweb.org/anthology/D19-1221
118. Wang A, Cho K (2019) BERT has a mouth, and it must speak: BERT as a Markov random field language model. In: Proc. of NeuralGen, pp 30–36, https://doi.org/10.18653/v1/W19-2304
119. Wang P, Qian Y, Soong FK, He L, Zhao H (2015) Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. arXiv preprint arXiv:1510.06168
120. Wang Y, Che W, Guo J, Liu Y, Liu T (2019) Cross-lingual BERT transformation for zero-shot dependency parsing. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference onnatural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp 5721–5727. https://doi.org/10.18653/v1/D19-1575. https://www.aclweb.org/anthology/D19-1575
121. Williams A, Nangia N, Bowman SR (2017) A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426
122. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al. (2016) Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144
123. Xing C, Wang D, Liu C, Lin Y (2015) Normalized word embedding and orthogonal transform for bilingual word translation. In: Proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Denver, Colorado, pp 1006–1011. https://doi.org/10.3115/v1/N15-1104. https://www.aclweb.org/anthology/N15-1104

124. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237
125. Yijia l (2019) Sentence-level language analysis with contextualized word embeddings. Ph.D. thesis, Harbin Institute of Technology
126. Zhang H, Gong Y, Yan Y, Duan N, Xu J, Wang J, Gong M, Zhou M (2019a) Pretraining-based natural language generation for text summarization. arXiv preprint arXiv:1902.09243
127. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019b) Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675
128. Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q (2019c) ERNIE: enhanced language representation with informative entities. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 1441–1451. https://doi.org/10.18653/v1/P19-1139. https://www.aclweb.org/anthology/P19-1139
129. Zhou J, Xu W (2015) End-to-end learning of semantic role labeling using recurrent neural networks. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, vol 1 (long papers). Association for Computational Linguistics, Beijing, China, pp 1127–1137. https://doi.org/10.3115/v1/P15-1109. https://www.aclweb.org/anthology/P15-1109
130. Zou WY, Socher R, Cer D, Manning CD (2013) Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the 2013 conference on empirical methods in natural language processing. Association for Computational Linguistics, Seattle, Washington, USA, pp 1393–1398. https://www.aclweb.org/anthology/D13-1141