Representing Words in Vector Space and Beyond

Benyou Wang, Emanuele Di Buccio, and Massimo Melucci

Abstract Representing words, the basic units in language, is one of the most fundamental concerns in Information Retrieval, Natural Language Processing (NLP), and related fields. In this paper, we reviewed most of the approaches of word representation in vector space (especially state-of-the-art word embedding) and their related downstream applications. The limitations, trends and their connection to traditional vector space based approaches are also discussed.

Keywords Word representation · Word embedding · Vector space

1 Introduction

This volume illustrates how quantum-like models can be exploited in Information Retrieval (IR) and other decision making processes. IR is a special and important instance of decision making because, when searching for information, the users of a retrieval system express their information needs through behavior (e.g., click-through activity) or queries (e.g., natural language phrases), whereas a computer system decides about the relevance of documents to the user's information need. By nature, IR is inherently an interactive activity which is performed by a user accessing the collections managed by a system through very interactive devices. These devices are immersed in a highly dynamic context where not only does the user's queries rapidly evolve but the collections of documents such as news or magazine articles also use words with different meanings. The main link between the "quantumness" of these models and IR is established by the vector spaces, which have for a long time been utilized to design modern computerized systems such as the search engines and they are currently the foundation of the most advanced methods for searching for multimedia information.

B. Wang (✉) · E. Di Buccio · M. Melucci
Department of Information Engineering, University of Padova, Padova, Italy
e-mail: wang@dei.unipd.it; dibuccio@dei.unipd.it; massimo.melucci@unipd.it

© Springer Nature Switzerland AG 2019

D. Aerts et al. (eds.), *Quantum-Like Models for Information Retrieval and Decision-Making*, STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics & Health, https://doi.org/10.1007/978-3-030-25913-6_5

Whatever the mathematical model or the retrieval function, documents and queries are mathematically represented as elements of sets, while the sets are labeled by words or other document properties. Queries, which are the most used data for expressing information needs, are sets or sequences of words or they are sentences expressed in a natural language; queries are oftentimes very short (e.g., one word) or occasionally much longer (e.g., a text paragraph). It is a matter of fact that the Boolean models for IR by definition view words as document sets and answer search queries with document sets obtained by set operators; moreover, the probabilistic models are all inspired to the Kolmogorov theory of probability, which is related to Boole's theory of sets; in addition, the traditional retrieval models based on vector spaces are eventually a means to provide a ranking or a measure to sets because they assign a weight to words and then to documents in the sets labeled by the occurring words. The implementation of content representation in terms of keywords and posting lists reflects the view of words as sets of documents and the view of retrieval operations as set operators. In this chapter, we will explain that a document collection can be searched by vectors embedding different words together, instead of by distinct words, by using the ultimate logic of *vector spaces*, instead of sets.

Representing words is fundamental for tasks which involve sentences and documents. Word embedding is a family of techniques that has recently gained a great deal of attention and aims at learning vector representation of words that can be used in these tasks. Generally speaking, embedding mainly consists in adopting a mapping, in which a fixed-length vector is typically used to encode and represent an entity, e.g., word, document, or a graph. Technically, in order to embed an object X in another object Y, the embedding is an injective and structure-preserving map $f : X \rightarrow Y$, e.g., user/item embedding [6] in item recommendation, network embedding [23], feature embedding in manifold learning [89], and word embedding. In this chapter, we will focus on word embedding techniques, which embed words in a low-dimensional vector space.

Word embedding is driven by the *Distributional Hypothesis* [33, 38], which assumes that linguistic items which occur in similar contexts should have similar meanings. Methods for modeling the distributional hypothesis can be mainly divided into the following categories:

- Vector-space models in Information Retrieval, e.g., [121], or representation in Semantic Spaces [67]
- Cluster-based distributional representation [17, 63, 79]
- Dimensionality reduction (matrix factorization) for document-word/word-word/word-context co-occurring matrix, also known as Latent Semantic Analysis (LSA) [24]
- Prediction based word embedding, e.g., using neural network-based approaches.

LSA was proposed to extract descriptors that capture word and document relationships within one single model [24]. In practice, LSA is an application of Singular Value Decomposition (SVD) to a document-term matrix. Following LSA, Latent Dirichlet Allocation (LDA) aims at automatically discovering the main topics

in a document corpus. A corpus is usually modeled as a probability distribution over a shared set of topics; these topics in turn are probability distributions over words, and each word in a document is generated by the topics [12]. This paper focuses on the geometry provided by vector spaces, yet is also linked to topic models, since a probability distribution over documents or features is defined in a vector space, the latter being a core concept of the quantum mechanical framework applied to IR [68, 69, 110].

With the development of computing ability for exploiting large labeled data, neural network-based word embedding tends to be more and more dominant, e.g., Computer Vision (CV) and Natural Language Processing. In the NLP field, neural network-based word embedding was firstly investigated by Bengio et al. [7] and further developed by [21, 75]. Word2vec [70][1] adopts a more efficient way to train word embedding, by removing non-linear layers and other tricks, e.g., hierarchical softmax and negative sampling. In [70] the authors also discussed the *additive compositional structure*, which denotes that word meanings can be composited with the addition of their corresponding vectors. For example, $\boldsymbol{king} - \boldsymbol{man} = \boldsymbol{queen} - \boldsymbol{women} = \boldsymbol{royal}$. This capability of capturing relationships among words was further discussed in [35] where a theoretical justification was provided. More importantly, Mikolov et al. [70] published open-source well-trained general word vectors, which made word embedding easy to use in various tasks.

In order to intuitively show the word vectors, some selected words (52 words about animals and 110 words about colors) are visualized in a 2-dimensional plane (as shown in Fig. 1) from one of the most popular Glove word vectors,[2] in which the position of the word is according to the reduced vector through a dimension reduction approach called T-SNE. It is shown that all the words are nearly clustered into two groups about colors and animals, respectively. For example, the word vectors of "rat" and "dog" are close to the word "cat," which is intuitively consistent to the Distributional Hypothesis since they ("cat" and "rat," or "cat" and "dog") may co-occur together with high frequencies.

Word embedding provides a more flexible and fine-grained way to capture the semantics of words, as well as to model the semantic composition of bigger-granularity units, e.g., from words to sentences or documents [71]. Some applications of word embedding will be discussed in Sect. 3. Although word embedding techniques and related neural network approaches have been successfully used in different IR and NLP tasks, they have some limitations, e.g., the polysemy and out-of-vocabulary problems. These issues have motivated further research in word embedding; Sect. 4.2 will discuss some of the current trends in word embedding that aim at addressing these issues. Moreover, we will discuss the link between the word vector representations and state-of-the-art approaches in modeling thematic structures.

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

[2] The words vectors are downloaded from http://nlp.stanford.edu/data/glove.6B.zip, with 6B tokens, 400K uncased words, and 50-dimensional vectors.

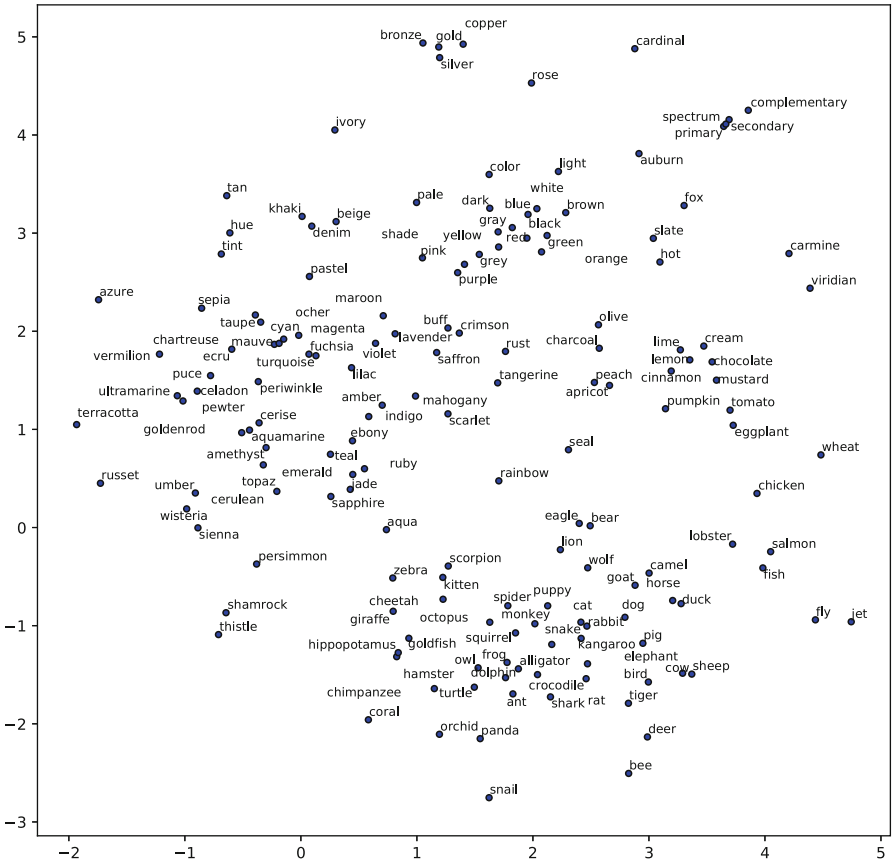

Fig. 1 The visualization of some selected words

2 Background

2.1 Distributional Hypothesis

Word embedding is driven by the *Distributional Hypothesis* [38]. The core of distributional hypothesis states that linguistic items with similar distributions have similar meanings and hence words with similar distributions should have similar representations. The distributional property is usually induced from document or textual neighborhoods (like sliding windows).

Some of the methods relying on the Distributional Hypothesis and the basic idea underlying them are reported below:

- Language model $p(w_k|w_{k-t}, w_{k-t+1}, \ldots w_{k-1})$: predicts the current word using previous words [7].
- Sequential scoring $p(w_{k-t}, w_{k-t+1}, \ldots w_k)$: predicts whether the given sentence is a legal one [21].
- Skip-gram $p(w_k|\ \forall w_i \in \{w_i|\ abs(k-i) < t\}\ )$: predicts a co-occurring word for each word [70].
- CBOW $p(w_k|w_{k-t}, w_{k-t+1}, \ldots w_{k-1}, \ldots w_{k+t})$: predicts a target word with context words (both previous ones and following ones) [70].
- Glove $p(\#(w_i, w_j)_{window}|w_i, w_j)$: predicts the co-occurring count between a word pair [78].

2.2 A Brief History of Word Embedding

While the *Distributional Hypothesis* was proposed many decades ago, the techniques of word embedding trained in a neural network has a much shorter history of about one and half decades [7], as mentioned in Sect. 1. Some typical ways to generate word vectors are introduced below.

NNLM The Neural Network Language Model (NNLM) [7] preliminarily aims to build a language model, while learning word embedding is not the main target. However, this is the first work in learning word vectors in a neural network (Fig. 2).

C&W The Collobert and Weston (C&W) approach was proposed in [21] in order to predict the fluency of a given sequence—see Fig. 3. One of the tasks in [21] assigns language modeling as a simple binary classification task: "if the word in the middle of the input window is related to its context or not" [21].

Skip-Gram Skip-gram balances a trade-off between performance and simplicity. As shown in Fig. 4, Skip-gram uses a word to predict one of its neighboring words.

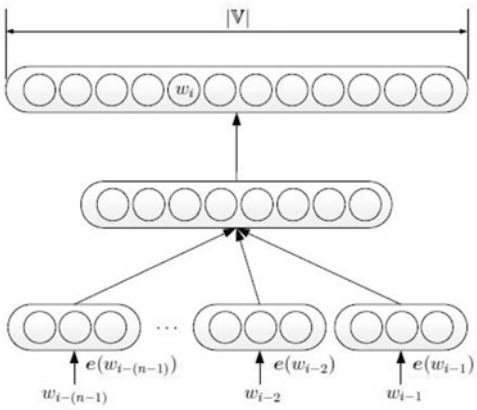

Fig. 2 NNLM concatenates all the word vectors in a sentence and then predicts the next word. → refers to the information flow in the forward neural network, while the circle denotes the neurons in the network. $|V|$ is the size of the word vocabulary [58]

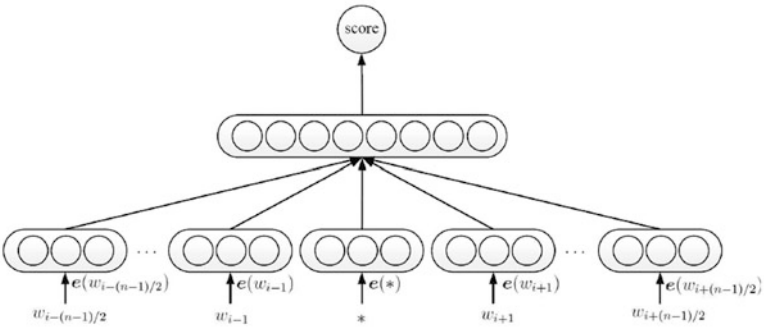

Fig. 3 C&W concatenates all the word vectors to predict whether it is a natural sentence or if it has replaced the center word with a random word [58]

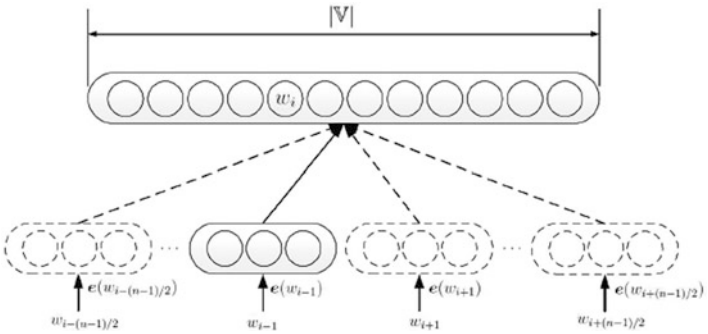

Fig. 4 Skip-gram directly uses one word to predict its neighboring word [58]

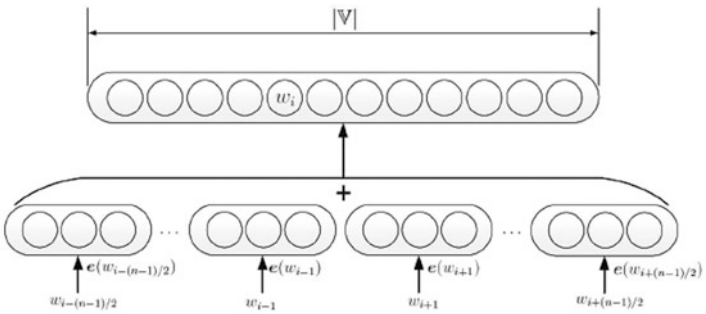

Fig. 5 CBOW uses the average embedding of the contextual words to predict the target word, where the contextual words are surrounded by the target word [58]

CBOW As shown in Fig. 5, CBOW uses context words to predict the current word. The difference between Skip-gram and CBOW is that in order to predict the target word, CBOW uses many words as the context, while Skip-gram uses only one neighboring word.

Glove Another popular word embedding named Glove[3] [78] takes advantage of global matrix factorization and local context window methods. It is worth mentioning that [60] explains that the Skip-gram with negative sampling derives the same optimal solution as matrix (Point-wise Mutual Information (PMI)) factorization.

3 Applications of Word Embedding

According to the input and output objects, we will discuss word-level applications in Sect. 3.1, sentence-level applications in Sect. 3.2, pair-level applications in Sect. 3.3, and seq2seq generation applications in Sect. 3.4. These applications can be the benchmarks to evaluate the quality of word embedding, as introduced in Sect. 3.5.

3.1 Word-Level Applications

Based on the learned word vector from a large-scale corpus, the word-level property can be inferred. Regarding *single-word level property*, word sentiment polarity is one of the typical properties. Word-pair properties are more common tasks, like word similarity and word analogy.

The advantage of word embedding is that: all the words, even from a complicated hierarchical structure like WordNet [31],[4] are embedded in a single word vector, thus leading to a very simple data structure and easy incorporation with a downstream neural network. Meanwhile, this simple data structure, namely a word-vector mapping, also provides some potential to share different knowledge from various domains.

3.2 Sentence-Level Application

Regarding sentence-level applications, the two typical tasks are sentence classification and sequential labeling, depending on how many labels the task needs. For a given sentence, there is only one final label for the whole sentence for text classification, where the number of labels in the sequential labeling is related to the number of tokens in the sentence (Fig. 6).

[3] https://nlp.stanford.edu/projects/glove/.

[4] An example of hierarchical structures is shown at the following address: http://people.csail.mit.edu/torralba/research/LabelMe/wordnet/test.html.

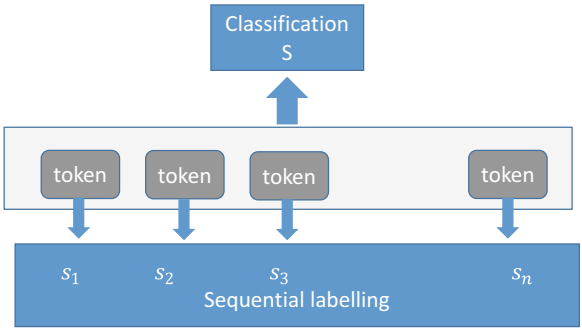

Fig. 6 Sentence-level applications: sentence classification and sequential labeling

Sentence Classification Sentence classification aims to predict the possible label for a given sentence, where the label can be related to the topic, the sentimental polarity, or whether the mail is spam. Text classifications were previously overviewed by Zhai [1], who mainly discussed the traditional textual representation. To some extent, trained word embedding from a large-scale *external* corpus (like Wikipedia pages or online news) is commonly used in IR and NLP tasks like text classification. Especially for a task with limited labeled data, in which it is impossible to train effective word vectors (usually with one hundred thousand parameters that need to be trained) due to the limited corpus, pre-trained embedding from a large-scale external corpus could provide general features. For example, average embedding (or with a weighted scheme) could be a baseline for many sentence representations and even document representations. However, due to the original error for the embedding training process in the external corpus and the possible domain difference between the current dataset and external corpus, adopting the embedding as features usually will not achieve significant improvement over traditional bag-of-word models, e.g., BM25 [88].

In order to solve this problem, the word vectors trained from a large-scale external corpus are only adopted as the initial value for the downstream task [51]. Generally speaking, all the parameters of the neural network are trained from scratch with a random or regularized initialization. However, the scale of the parameter in the neural network is large and the training samples may be small. Moreover, the trained knowledge from another corpus is expected to be used in a new task, which is commonly used in Computer Vision (CV) [41]. In an extreme situation, the current dataset is large enough to implicitly train the word embedding from scratch; thus, the effect of pre-initial embedding could be of little importance.

Firstly, multi-layer perception is adopted over the embedding layers. Kim et al. [51] first proposed a CNN-based neural network for sentence classification as shown in Fig. 7. The other typical neural networks named Recurrent Neural Network (and its variant called Long and Short Term Memory (LSTM) network [43] as shown in Fig. 8) and Recursive Neural Network [36, 81], which naturally process sequential sentences and tree-based sentences, are becoming more and more popular. In particular, word embedding with LSTM encoder–decoder architecture

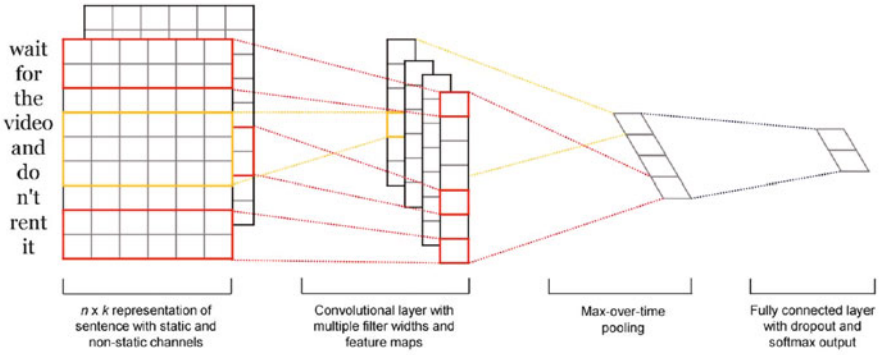

Fig. 7 CNN for sentence modeling [52] with convolution structures and max pooling

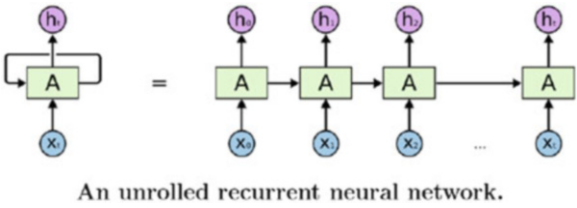

Fig. 8 LSTM. The left subfigure shows a recurrent structure, while the right one is unfolded over time

[3, 18] outperformed the classical statistic machine translation,[5] which dominates machine translation approaches. Currently, the industrial community like Google adopts completely neural machine translation and abandons statistical machine translation.[6]

Sequential Labeling Sequence labeling aims to classify each item of a sequence of observed value, with the consideration of the whole context. For example, Part-Of-Speech (POS) tagging, also called word-category disambiguation, is the process of assignment of each word in a text (corpus) to a particular part-of-speech label (e.g., noun and verb) based on its context, i.e., its relationship with adjacent and related words in a phrase or sentence. Similar to the POS tagging, the segment tasks like Named Entity Recognition (NER) and word segment can also be implemented in a general sequential labeling task, with definitions of some labels like begin label (usually named "B"), intermediate label (usually named "O"), and end label (usually named "E"). The typical architecture for sequence labeling is called BiLSTM-CRF [46, 59], which is based on bidirectional LSTMs and conditional random fields, as shown in Fig. 9.

[5] http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf.

[6] https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/.

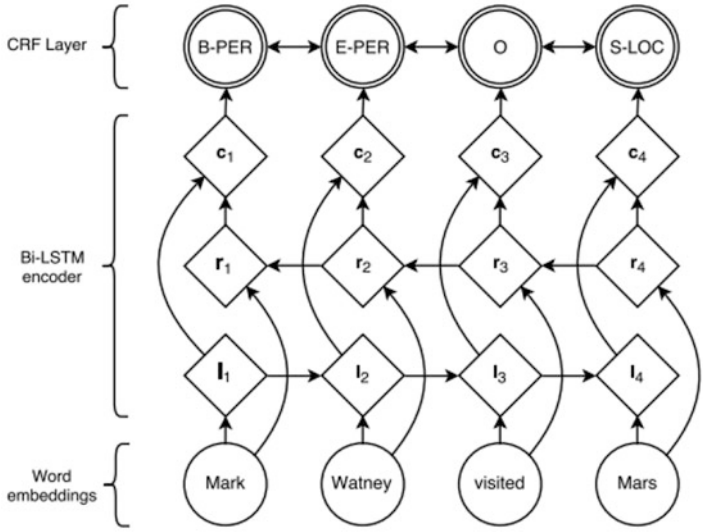

Fig. 9 LSTM-CRF for named entity recognition [59]

Document-Level Representation Similar to the methods for sentence-level representation, a document with mostly multiple sentences, which can also be considered a long "sentence," needs an adaption for more tokens. A document mostly consists in multiple sentences. If we interpret a document as a long sentence, we can use the same approaches proposed for the sentence-level applications while taking into account the fact that there are more tokens. For example, a hierarchical architecture is usually adopted for document representation, especially in RNN, as shown in Fig. 10. Generally speaking, all the sentence-level approaches can be used in document-level representation, especially if the document is not so long.

3.3 Sentence-Pair Level Application

The difference between sentence applications and sentence-pair applications is the extra *interaction* module (we call it a matching module), as shown in Fig. 11. Evaluating the relationship between two sentences (or a sentence pair) is typically considered a matching task, e.g., information retrieval [73, 74, 129], natural language inference [14], paraphrase identification [27], and question answering. It is worth mentioning that the Reading Comprehension (RC) task can also be a matching task (especially question answering) when using an extra context, i.e., a passage for background knowledge, while the question answering (answer selection) does not have specific context. In the next subsection, we will introduce the Question Answering task and Reading Comprehension task.

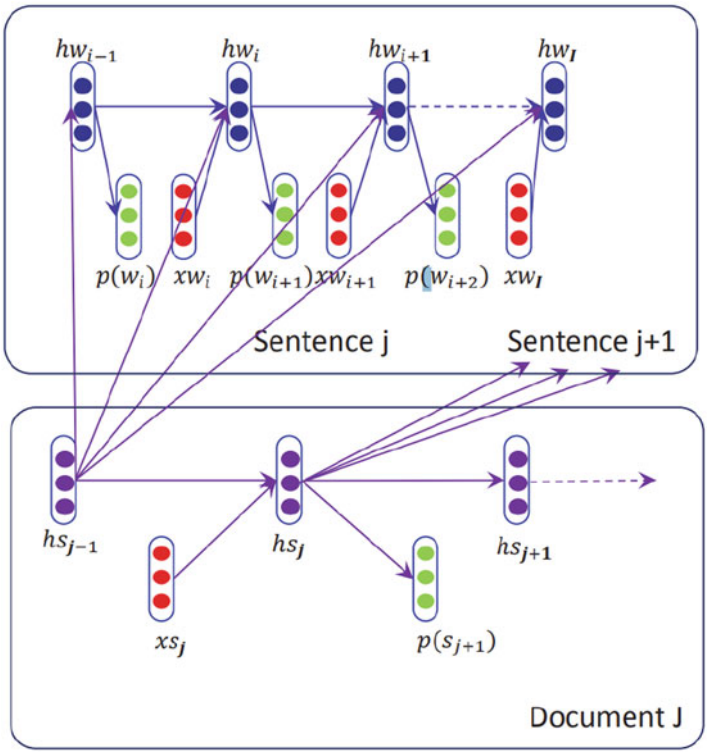

Fig. 10 Hierarchical recurrent neural network [64]

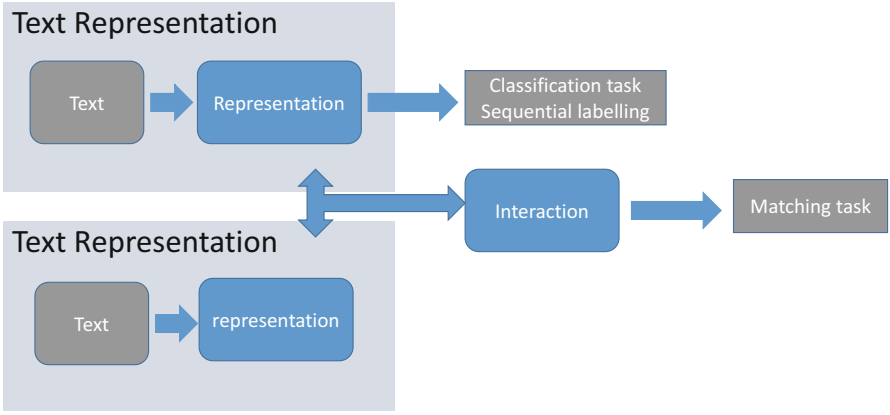

Fig. 11 The figure shows that the main difference between a sentence-pair task and a sentence-based task is that there is one extra interaction for the matching task

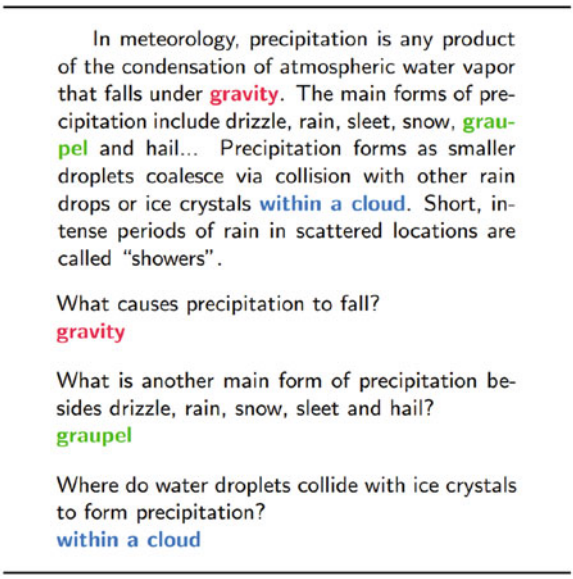

Fig. 12 A demo of SQuAD dataset [85]

Question Answering Differently from *expert system*s with structured knowledge, question answering in IR is more about retrieval and ranking tasks in limited unstructured document candidates. In some literature, reading comprehension is also considered a question answering task like SQuAD QA. Generally speaking, reading comprehension is a question answering task in a specific context like a long document with some internal phrases or sentences as answers, as shown in Fig. 12. Table 1 reports current popular QA datasets.

In order to compare the neural matching model and non-neural models, we focus on TREC (answer selection), which has limited answer candidates, instead of an unstructured document as context in reading comprehension. Some matching methods are shown in Table 2, which mainly refers to the ACL wiki page.[7]

3.4 Seq2seq Application

Seq2seq is a kind of task with both input and output as sequential objects, like a machine translation task. It mainly uses an encoder–decoder architecture [19, 100] and further attention mechanisms [3], as shown in Fig. 13. Both the encoder and decoder can be implemented as RNN [19], CNN [34], or only attention mechanisms (i.e., Transformer [111]).

[7] https://aclweb.org/aclwiki/Question_Answering_(State_of_the_art).

Table 1 Popular QA dataset

Dataset	Characteristics	Main institution	Venue
TREC QA [119][a]	Open-domain question answering	CMU	EMNLP 2007
Insurance QA [32]	Question answering for insurance	IBM Watson	ASRU 2015
Wiki QA [123]	Open-domain question answering	MS	EMNLP 2015
Narrative QA [53]	Reading Comprehension	DeepMind	TACL 2018
SQuAD 1.0 [85]	Questions for machine comprehension	Standford	EMNLP 2016
MS Marco [76]	Human-generated machine reading	MS.	NIPS 2016
NewsQA [107, 108]	Reading comprehension	Maluuba	RepL4NLP 2017
TriviaQA [48]	Reading comprehension distantly supervised labels	Allen AI	ACL 2017
SQA [47]	Sequential question answering	U. of Maryland & MS.	ACL 2017
CQA [102]	QA with knowledge base of web	Tel-Aviv university	NAACL 2018
CSQA [92]	Complex sequential QA	IBM	AAAI 2018
QUAC [20][b]	Question answering in context	Allen AI	EMNLP 2018
SQuAD 2.0 [84]	SQuAD with unanswered questions	Standford	ACL 2018
CoQA [87][c]	Conversational question answering	Standford	Aug. 2018
Natural questions [57]	Natural questions in Google search	Google	TACL 2019

The frequent publishing of QA datasets demonstrates that the academic community is paying more and more attention to this task. Almost all the researchers in this community tend to use word embedding-based neural networks for this task

[a] http://cs.stanford.edu/people/mengqiu/data/qg-emnlp07-data.tgz

[b] http://quac.ai/

[c] https://stanfordnlp.github.io/coqa/

3.5 Evaluation

The basic evaluations of word embedding techniques are based on the above applications [94], e.g., word-level evaluation and downstream NLP tasks like those mentioned in the last section, as shown in [58]. Especially for a downstream task, there are two common ways to use word embedding, namely as fixed features or by treating it only as initial weights and fine-tuning it. We mainly divide it into two part of evaluations, i.e., context-free word properties and embedding-based downstream NLP tasks, while the latter may involve the context and the embedding can be fine-tuned.

Word Property Examples of the context-free word properties include word polarity classification, word similarity, word analogy, and recognition of synonyms and antonyms. In particular, one of the typical tasks is called an analogy task [70], which

Table 2 State-of-the-art methods for sentence selection, where the evaluation relies on the TREC QA dataset

Algorithm	Reference	MAP	MRR
Mapping dependencies trees [82]	AI and math Symposium 2004	0.419	0.494
Dependency relation [22]	SIGIR 2005	0.427	0.526
Quasi-synchronous grammar [119]	EMNLP 2007	0.603	0.685
Tree edit models [42]	NAACL 2010	0.609	0.692
Probabilistic tree edit models [118]	COLING 2010	0.595	0.695
Tree edit distance [124]	NAACL 2013	0.631	0.748
Question classifier, NER, and tree kernels [95]	EMNLP 2013	0.678	0.736
Enhanced lexical semantic models [126]	ACL 2013	0.709	0.770
DL with bigram+count [128]	NIPS 2014 DL workshop	0.711	0.785
LSTM—three-layer BLSTM+BM25 [116]	ACL 2015	0.713	0.791
Architecture-II [32, 45]	NIPS 2014	0.711	0.800
L2R + CNN + overlap [96]	SIGIR 2015	0.746	0.808
aNMM: [122] attention-based neural matching model	CIKM 2016	0.750	0.811
Holographic dual LSTM architecture [104]	SIGIR 2017	0.750	0.815
Pairwise word interaction modeling [40]	NAACL 2016	0.758	0.822
Multi-perspective CNN [39]	EMNLP 2015	0.762	0.830
HyperQA (hyperbolic embeddings) [103]	WSDM 2018	0.770	0.825
PairwiseRank + multi-perspective CNN [86]	CIKM 2016	0.780	0.834
BiMPM [120]	IJCAI 2017	0.802	0.875
Compare-aggregate [8]	CIKM 2017	0.821	0.899
IWAN [97]	EMNLP 2017	0.822	0.889
IWAN + sCARNN [106]	NAACL 2018	0.829	0.875
NNQLM [131]	AAAI 2018	0.759	0.825
Multi-cast attention networks (MCAN) [105]	KDD 2018	0.838	0.904

Recent papers about TREC QA used embedding-based neural network approaches, while previous ones were based on some traditional methods like IR approaches and edit distance

mainly targets both the syntactic and semantic analogies. For instance, "man is to woman" is semantically similar to "king is to queen," while "predict is to predicting" is syntactically similar to "dance is to dancing." Word Embedding methods achieve good performance in the above word-level tasks, which demonstrates that the word embedding can capture the basic semantic and syntactic properties of the word.

Downstream Task If word embedding is used in a context, which means we consider each word in a phrase or sentence for a specific target, we can train the word embedding by using the labels of the specific task, e.g., sequential labeling, text classification, text matching, and machine translation. These tasks are divided by the pattern of input and output, shown in Table 3.

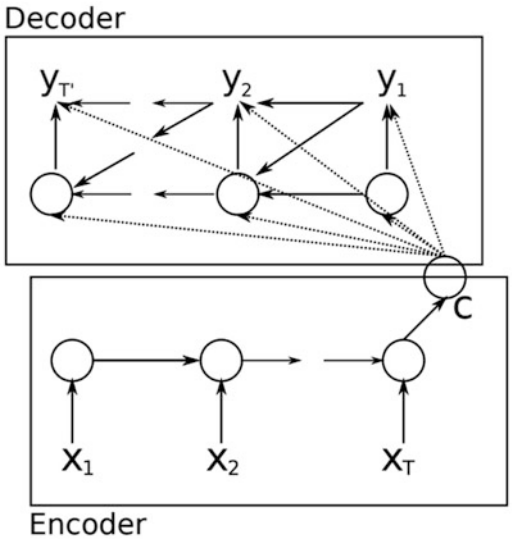

Fig. 13 An illustration of the proposed Seq2seq (RNN Encoder–Decoder)

Table 3 The difference of the downstream tasks

Algorithm	Input	Output	Typical tasks	Typical models
Text classification	S	$\mathcal{R}$	Sentiment analysis, topic classification	Fastext/CNN/RNN
Text matching	(S_1, S_2)	$\mathcal{R}$	QA, reading comprehension	aNMM,DSSM
Sequential labeling	S	$\mathcal{R}^{\|S\|}$	POS, word segmentation, NER	LSTM-CRF
Seq2Seq	S_1	S_2	machine translation, abstraction	LSTM/Transformer encoder–decoder

Generally speaking, the tasks for the word properties can partially reflect the quality of the word embedding. However, the final performance in the downstream tasks may vary. It is more reasonable to directly assess it in the real-world downstream tasks as shown in Table 3.

4 Reconsidering Word Embedding

Some limitations and trends of word embedding are introduced in Sects. 4.1 and 4.2. We also try to discuss the connections between word embedding and topic models in Sect. 4.3. In Sect. 4.4, the dynamic properties of word embedding are discussed in detail.

4.1 Limitations

Limitation of Distributional Hypothesis The first concern directly targets the effectiveness of the distributional hypothesis. Lucy and Gauthier [66] find that while word embeddings capture certain *conceptual* features such as "is edible" and "is a tool," they do not tend to capture *perceptual features* such as "is chewy" and "is curved," potentially because the latter are not easily inferred from distributional semantics alone.[8]

Lack of Theoretical Explanation Generally, humans perceive the words with various aspects other than only the semantic aspect, e.g., sentimental polarity and semantic hierarchy like WordNet. Thus, mapping a word to a real-valued vector is a practical but preliminary method, which leads to limited hints for humans to understand. For a given word vector, it is hard for humans to know what exactly the word means; the scalar value of each element in a word vector does not provide too much physical meaning. Consequently, it is difficult to interpret obtained vector space from the human point of view.

Polysemy Problem Another problem with word embeddings is that they do not account for *polysemy*, instead assigning exactly one vector per surface form. Several methods have been proposed to address this issue. For example, Athiwaratkun and Wilson [2] represent words not by single vectors, but by Gaussian probability distributions with multiple modes—thus capturing both uncertainty and polysemy. Upadhyay et al. [109] leverage multi-lingual parallel data to learn multi-sense word embeddings, for example, the English word "bank," which can be translated into both the French words *banc* and *banque* (evidence that "bank" is polysemous), and help distinguish its two meanings.

Out-Of-Vocabulary Problem With a pre-trained word embedding, some words may not be found in the vocabulary of the pre-trained word vectors, that is, the Out-Of-Vocabulary (OOV) problem. If there are many OOV words, the final performance decreases largely due to the fact that we use a partial initialization from the given word vectors, while other words are randomly initialized, instead of initializing all the weights. This happened more frequently in some professional domains, like medicine text analysis, since it is not easy to find some professional words in a general corpus like Wikipedia.

Semantic Change Over Time One of the limitations of most word embedding approaches is that they assume that the meaning of a word does not change over time. This assumption can be a limitation when considering corpora of historic texts or streams of text in newspapers or social media. Section 4.4 will discuss some recent works which aim to explicitly include the temporal dimensions in order to capture how the word meaning changes over time.

[8] http://www.abigailsee.com/2017/08/30/four-deep-learning-trends-from-acl-2017-part-1.html.

4.2 Trends

Interpretability One of the definitions of "interpretability" is proposed by Lipton [65]. In particular, Lipton [65] identifies two broad approaches to interpretability: *post-hoc explanations* and *transparency*. Post-hoc explanations take a learned model and draw some useful insights from it; typically these insights provide only a partial or **indirect** explanation of how the model works. The typical examples are visualization (e.g., in machine translation [26]) and transfer learning.

Transparency asks more **directly** "how does the model work?" and seeks to provide some way to understand the core mechanisms of the model itself. As Manning said, "Both language understanding and artificial intelligence require being able to understand bigger things from knowing about *smaller parts*."[9] Firstly, it is more reasonable to build a bottom-up system with linguistically structured representations like syntax or semantic parsers and sub-word structures (refer to Sect. 4.2) than an end-2-end system without consideration of any linguistic structures. Moreover, we can use some constrains to normalize each subcomponent and make it understandable for humans, as well as relieve the non-convergent problems. For instance, an attention mechanism [3] is one of the most successful mechanisms from the point view of normalization. For an unconstrained real-valued vector, it is hard to understand and know how it works. After the addition of a softmax operation, this vector denotes a multinomial probability distribution in which each element ranges from 0 to 1 and the sum of the vectors equals 1.

Contextualized Word Embedding Previously, word embedding was static, which means it did not depend on the context and it was one-to-one mapping from a word to a static vector. For example, the word "bank" has at least two meanings, i.e., "the land alongside or sloping down to a river or lake" and "a financial establishment that invests money deposited by customers, pays it out when required, makes loans at interest, and exchanges currency." However, the word in a finance-related context and a river-related context could be mapped into the same fixed vector, which is not reasonable for language. Instead of storing a static look-up table, contextualized word embedding learns a language model to generate a real-time word vector for each word based on the neighboring word (context). The first model was proposed with the name Embedding from Language MOdel (ELMO) [80], and it was further investigated by Generative Pre-Training (GPT) [83] and BERT [25]. More specifically, BERT obtained new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE benchmark, MultiNLI accuracy, and the SQuAD with huge improvements.

Linguistically Enhanced Word Embedding One of the main criticisms of word embedding is that it ignores the linguistic knowledge and instead adopts a brute force approach which is totally driven by data. However, there are already many

[9] https://nlp.stanford.edu/manning/talks/Simons-Institute-Manning-2017.pdf.

linguistic resources for words, e.g., WordNet and sentimental lexicon. Incorporation of the linguistic knowledge trends in the current paradigm of the NLP can relieve the dependence of data. These linguistic resources are expected to enhance the representative power of word embedding, which may be used in higher layers than word embedding layers like syntax structures [101] or only word embedding with WordNet and related lexicon resources [30, 61].

Sub-Word Embedding We briefly discussed the OOV problem in Sect. 4.1. Previous solutions for relieving it were commonly based on empirical insights, e.g., assigning a special token to all the OOV words. In [132] character-based embedding for text classification was adopted, avoiding directly processing the word-level embedding. In the sub-word embedding, there are no OOV problems since the proposed approaches directly build the word embedding with the units with smaller granularity which may have a limited number. For example, one of the sub-word approaches in English is based on characters, which are limited to a–z, A–Z, 0–9, punctuation, and other special symbols. Moreover, a character level approach could be beneficial for some specific languages, like Chinese, that can make use of smaller-granularity units which are smaller than words but also have abundant semantic information, like components. Sub-word regularization [54] trains the model with multiple sub-word segmentation (based on a unigram language model) probabilistically sampled during training. These works demonstrate that there is some potential to incorporate some fine-refined linguistic knowledge in the neural network [13, 54].

Advanced Word Embedding: Beyond one Fixed Real-Valued Vector More recently, different types of word embedding beyond real-valued vectors have been developed, for example:

- **Gaussian embedding** [112] assigns a Gaussian distribution for each word, instead of a point vector in a low-dimension space. The advantages are that it naturally captures uncertainty and expresses asymmetries in the relationship between two words.
- **Hyperbolic embedding** [77, 93] embeds words as points in a Cartesian product of hyperbolic spaces; therefore, the hyperbolic distance between two points becomes the Fisher distance between the corresponding probability distribution functions (PDFs). This additionally derives a novel principled "is-a" score on top of word embeddings that can be leveraged for hypernymy detection.
- **Meta embedding** [50, 127] adopts multiple groups of word vectors and adaptively obtains a word vector by leveraging all the word embeddings.
- **Complex-valued embedding** [62, 114] formulates a linguistic unit as a complex-valued vector, and links its length and direction to different physical meanings: the length represents the relative weight of the word, while the direction is viewed as a superposition state. The superposition state is further represented in an amplitude-phase manner, with amplitudes corresponding to the lexical meaning and phases implicitly reflecting the higher-level semantic aspects such as polarity, ambiguity, or emotion.

4.3 Linking Word Embedding to Vector-Space Based Approaches and Representation of Thematic Structures

Deriving the Topic Distribution from Word Embedding Research on the representation of themes in an unstructured document corpus—finding word patterns in a document collection—dates back to the 1990s, i.e., to the introduction of LSA [24]. A subsequent extension that exploits a statistical model was proposed By Hofmann in[44]. That model, named Probabilistic Latent Semantic Indexing (PLSI), relies on the *aspect model*, a latent variable model for co-occurrence data where an occurrence—in our case a word occurrence—is associated with an unobserved/latent variable. The work by Hofmann and subsequent works rely on the "same fundamental idea—that a document is a mixture of topics—but make slightly different statistical assumptions" [99]. For instance, in [12] Blei et al. extended the work by Hofmann making an assumption on how the mixture weights for the topics in a document are generated, introducing a Dirichlet prior. This line of research is known as *topic modeling*, where a topic is interpreted as a group of semantically related words. Since the focus of this paper is not on topic modeling, in the remainder of this section we are going to introduce only the basic concepts needed to discuss possible links with word embedding approaches; the reader can refer to the work reported in [9, 11, 15, 99] for a more comprehensive discussion on the difference among the diverse topic models and the research trends and direction in topic modeling.

As mentioned above, probabilistic topic models consider the document as a distribution over topics, while the topic is a distribution over words. In PLSI no prior distributions are adopted and the joint probability distribution between document and word is expressed as follows:

$$p(w,d) = \sum_{c \in C} p(w,d,c) = p(d) \sum_{c \in C} p(c,w|d) = p(d) \sum_{c \in C} p(c|d)p(w|c), \tag{1}$$

where d is a document, while w is a specific word and C is the collection of topics. A crucial point of topic models is how to estimate the $p(c|d)$ and $p(w|c)$.

Using an "empirical" approach, we can also get the $p(c|d)$ and $p(w|c)$ from word embedding. Suppose that we obtain a word embedding, i.e., a mapping from a word (denoted as an index with a natural number) to a dense vector $\mathcal{N} \rightarrow \mathcal{R}^n$. For a given sentence S with words sequence $\{w_1, w_2, \ldots w_n\}$, we can get a representation for s with an average embedding like [49], namely $\boldsymbol{d} = \sum_{i=1}^{n} \boldsymbol{w_i}$. It is easy to define a topic with distribution $p(w|c)$, represented as: $\boldsymbol{c_j} = \sum_{i=1}^{|V|} p_{w_i|c} \boldsymbol{w_i}$, $\boldsymbol{c_j} \in C$. Then we can obtain the following topic distribution of a document:

$$p(c_j|d) = \frac{e^{-||\boldsymbol{d}-\boldsymbol{c_j}||_2}}{\sum_i^{|C|} e^{-||\boldsymbol{d}-\boldsymbol{c_i}||_2}}. \tag{2}$$

The relationship between word embedding and topic models has been addressed in the literature. For instance, the work reported in [60] shows that a special case of word embedding, i.e., Skip-gram, has the same optimal solution as the factorization of a shifted Point-wise Mutual Information (PMI) matrix.[10] Empirically, the count-based representations and distributed representations can be combined together with complementary benefits [78, 115].

Recent works focused on exploiting both methods. The discussion of previous approaches reported in [98] reports on two lines of research: methods used to improve word embedding through the adoption of topic models, which addresses the polysemy problem; methods used to improve topic models through word embedding, which obtains more coherent words among the top words associated with a topic. These approaches mainly rely on a pipeline strategy, "where either a standard word embedding is used to improve a topic model or a standard topic model is used to learn better word embeddings" [98]. The limitation of these approaches is the lack of capability to exploit the mutual strengthening between the two, which a joint learning strategy, in principle, could exploit. This is the basic intuition underlying the work reported in [98]. Another example is *lda2vec* where the basic idea was "modifying the Skip-gram Negative-Sampling objective in [71] to utilize document-wide feature vectors while simultaneously learning continuous document weights loading onto topic vectors." The work reported in [117] proposes a different approach relying on a "topic-aware convolutional architecture" and a reinforcement learning algorithm in order to address the task of text summarization.

Regarding the Contextual Windows The previous subsection suggests possible connections between word embedding and the representation of the thematic structure in document corpora, e.g., through topic models. Vector space based approaches in IR, topic models, matrix factorization, and word embedding can be considered as different approaches relying on distributional hypothesis as discussed in Sect. 1. One of the differences among these methods may be how to choose the size of the contextual window. In this paper, we classify the contextual window into several sizes, i.e., "character → word → phase/N-gram → clause → sentence → paragraph → document," ordered from the smallest to the biggest granularity. For example, Vector Space Model (VSM) in IR usually chooses the whole document as the context; thus, it may capture the document-level feature of text, like the thematic structure. Approaches based on word-word matrix factorization usually set a smaller window size to statistically analyze the co-occurrence between words—similar to the windows of CBOW [70], thus targeting a smaller context in order to capture the word-level feature related to its word meaning.

Depending on the context size, features in vector space based approaches in IR are already at a relatively high level, e.g., the TFIDF vector or the language model [130], and they can be used directly for relatively downstream task like

[10]The shifted PMI matrix is "the well-known word-context PMI matrix from the word-similarity literature, shifted by a constant offset" [60].

document ranking. Lower-level word features of word-word matrix factorization (or CBOW) can be used directly for the relatively upstream task like morphology, lexicon, and syntax, and it needs some abstraction components to extract from the low-level features to high-level features. On the other hand, abstraction from the low-level features to high-level features may imply a loss of some fundamental lexical meaning. The low-level features (word-word matrix factorization or CBOW) are usually considered a better basic input for another "stronger" learning model—e.g., when using multiple layers of non-linear abstraction—compared to higher-level features.

4.4 Towards Dynamic Word Embedding

One of the limitations of most representations of words, documents, and themes is that they do not consider the *temporal dimension*. This is crucial when considering corpora such as historical document archives, newspapers, or social media, e.g., tweets, that consist in a continuous stream of informative resources. The use of these "time-stamped" resources is useful not only for general tasks but also for specialist users. Indeed, the tasks performed by the specialists of a discipline need to make hypotheses from data, for example, by means of longitudinal studies. This is the case for the tasks performed by specialists in the field of Social Science, Humanities, Journalism, and Marketing.

Let us consider, for instance, the case of sociologists that study the public perception of science and technology by the public opinion—this line of research is known as STS, Science and Technology Studies. The study of how some science and technology-related issues are discussed by the media, e.g., newspapers, could be useful in providing policy makers with insights on the public perception of some issues on which they should or intend to take actions or provide guidance on the way these issues should be publicly discussed (e.g., on the use of "sensible" words or aspects related to the issues). In this context, relevant information can be gained from how the meaning of a word or how the perception of an issue related to a word change through time.

Previous works on topic modeling addressed the issue of including the temporal dimension, specifically, the issue that topics can change over time. In [72] Mimno proposes a possible approach to visualize the topic coverage across time starting from topic learnt using a "static" approach: given the probabilities and the topic assignment estimated via LDA, the topic trend can be visualized by counting the number of words in each topic published in a given year and then normalizing over the total number of words for that year. Other works embedded the time dependence directly in the statistical model. One of the earliest works is that proposed in [10] where dynamic topic models were introduced. The underlying assumption is that time is divided into time slices, e.g., by years; documents in a specific time slice are modeled using a K-component topic model—K is the number of topics—where topics in a given time slice evolve from those in the previous time slice. This kind

of representation could be extremely useful for a specialist in order to follow the evolution of a single word, e.g., by inspecting the top words for diverse topics where the word is framed in his research hypothesis—e.g., the "nuclear" word framed in "innovation," "risk," or "energy" topics—or following the posterior estimate of the frequency of the word as a function of the year, as shown in [10]. As stated by the authors, one of the limitations of that approach is that the number of topics needs to be specified beforehand; the work reported in [28] aimed to address this limitation by introducing a non-parametric version for modeling topics over time.

Even if dynamic/time-aware versions of topic models can support specialists in their investigation, the adoption of word embedding to study changes in a word representation could provide complementary evidence to support or undermine a research hypothesis. Indeed, as mentioned above, topic models are learned from a more "global view," while word embedding exploits a more "local view," e.g., using evidence from local context windows; this local view might help to obtain a word representation that, in a way, "reflects the semantic, and sometimes also syntactic, relationships between the words" [98]. Another point of view about the difference between topic models and word embedding approaches could be the scale of the dimension and the sparseness degree in the vector space. Intuitively, topic models (especially the topic distribution over words) tend to adopt sparse vectors with bigger dimensions, while the word embedding approaches adopt low-dimension dense vectors which may save some memory space and provide more flexibility for the high-level applications. Note that the difference in sparseness can be decreased to some extent by the sparsing regularization as introduced by Vorontsov et al. [113].

The work reported in [55] discussed several approaches to identify "linguistic change." As an example of linguistic change, they referred to the change of the word "gay" that shifted from the meaning of "cheerful" or "frolicsome" to homosexuality (see Fig. 1 of that paper). They proposed three different approaches to generate time series aimed to capture different aspects of word evolution across time: a frequency-based method, a syntactic method, and a distributional method. Because of the objective of this survey, we will focus on the last one. They divided the entire time span of the dataset in time slices of the same size, e.g., 1-month or 5-year slices. Then a word embedding technique—`gensim` implementation of the Skip-gram model—was used to learn word representation in each slice; an alignment procedure was then adopted to consider all the embeddings in a unique coordinate system. Finally, the time series was obtained by calculating the distance between the time 0 and the time t in the embedding space of the final time slice. The use of time series has several benefits, e.g., the possibility to use change point detection methods to identify the point in time where the new word meaning became predominant. The distributional approach was the most effective in the various evaluation settings: synthetic evaluation, evaluation on a reference dataset, and evaluation with human assessors.

In [37] the change in meaning of a word through time is referred to as a "semantic change." The authors report several examples in word meaning change, e.g., the semantic change of the word "gay" as in [55] and that of the word "broadcast," which at the present time is mainly intended as a synonym of "transmitting signal."

While in the early twentieth century it meant "casting out seeds." In that work, static versions of word embedding techniques were used, but word embedding was learned for each time slice and then aligned in order to make word vectors from different time periods comparable; aligning is addressed as an Orthogonal Procrustes Problem. Three word embedding techniques were considered. The first is based on Positive Point-wise Mutual Information (PPMI) representations, where PPMI values are computed with respect to pre-specified context words and are prepared in a matrix whose rows are the word vector representations. The second approach, in the paper referred to as SVD, considers a truncated version of the SVD of the PPMI matrix. The last method is Skip-gram with negative sampling. The work reported in that paper is pertinent to our "specialist user scenario" since the main contribution is actually a methodology to investigate two research hypotheses. In particular, the second hypothesis investigated is that "Polysemous words change at faster rates"; this is related to an old hypothesis in linguistics that dates back to [16] and states that "words become semantically extended by being used in diverse contexts." Subsequent works [29] show that the results obtained in the literature for diverse hypotheses on semantic change—including those in [37]—should be revised; using as a control test an artificially generated corpus with "no semantic change" as a control test, they showed that the previously proposed methodologies detected a semantic change in the control test as well. The same result was observed for diverse hypotheses—see the survey reported in [56] for an overview of the diverse hypotheses investigated. As mentioned by Dubossarsky et al. [29], their result supports further research in evaluation of dynamic approaches "articulating more stringent standards of proof and devising replicable control conditions for future research on language change based on distributional semantics representations" [29].

The work reported in [90] introduces a dynamic version of the exponential family of embedding previously proposed in [91]. The reason for the introduction of the exponential family of embedding was to generalize the idea of word embedding to other data, e.g., neuronal activity or shopping for an item on the basis of the context (other items in the shopping cart). The obtained results show that the dynamic version of the exponential family embedding provides better results in terms of conditional likelihood of held-out predictions when compared with static embeddings [71, 91] and time-binned embeddings [37].

In [4] the authors extend the Bayesian Skip-gram Model proposed in [5] to a dynamic version considering a diffusion process of the embedding vectors over time, more specifically a Ornstein–Uhlenbeck process. Both of the two proposed variants resulted in more smoothed word embedding trajectories[11] than the baselines, which utilized the approach proposed in [37].

In [125] the authors proposed to find temporal word embedding to solve a joint optimization problem where the "key" component is a smoothing term that

[11] Trajectories where based on the cosine distance between two words representation over time.

encourages embedding to be aligned, thus explicitly solving the alignment problem while learning embedding and avoiding a two-step strategy like that adopted in [37] or in [55].

In [56] the authors report a number of open issues concerning the study of temporal aspects of semantic shifts. Two challenges that are particularly relevant to the works reported in this chapter and this venue are: (1) *the lack of formal mathematical models of diachronic embeddings*; (2) the need for *robust gold standard test sets of semantic shifts*; (3) the need for *algorithms able to work on small datasets*. With regard to the first point, investigating quantum-inspired models could be a possible research direction to find a formal mathematical framework to model dynamic/diachronic word embeddings, e.g., exploiting the generalized view of probability and the theory of time evolution of systems. With regard to the second point, and evaluation in general, a possible direction is to devise tasks with specialists, e.g., journalists, linguists, or social scientists, to create adequate datasets. This is also related to the last point, i.e., the need for algorithms that are "robust" to the size of the dataset: indeed, specialists, even when performing longitudinal user studies, can rely on relatively small datasets in order to investigate specific research issues. On the basis of the ongoing collaboration with sociologists and linguists, another open issue that could be really beneficial for the specialists investigations is "identifying groups of words that shift together in correlated ways" [56]; this could be particularly useful to investigate how some thematic issues are perceived by the public opinion and how this perception varies through time. As suggested by the results reported in [29], evaluation protocols to measure these algorithms' effectiveness should be rigorously designed.

As mentioned above, word embedding and topic models are based on two very different views. Rudolph et al. [90] suggest another possible research direction in the dynamic representation of words: devise models able to combine the two approaches and exploit their "complementary" representations in dynamic settings.

5 Conclusion

We introduced many vector space based approaches for representing words, especially the word vector techniques. Regarding the word vector, we introduced many variants presented throughout in the history and their limitations and trends. A concise summary is reported in Table 4.

Since the effectiveness of word embedding is supported by the investigation in many NLP and IR tasks and by many benchmarks, it is worth investigating further. In the future, it is expected to incorporate some external knowledge like linguistic features or the common sense of humans (like knowledge base) to word vectors. Besides these empirical efforts, some theoretical understanding is also important to this field, like the interpretability about why it works and where it does not work.

Table 4 A summary including various word vector techniques

Algorithm	Polysemy	Interpretability	OOV	Speed
NNLM [7]				
C&W [21]				
Skip-gram [70]				+
CBOW [70]				+
Glove [78]		+		
Char-based embedding [132]			+	
Elmo [80]	+			
BERT [25]	+			
Gaussian embedding [112]	+			
Hyperbolic embedding [77]		+		
Meta embedding [127]	+			
Complex embedding [62]		+		

Some earlier works aim to develop fast-training methods, while recent works focus more on the empirical performance with dynamic context-aware embedding and the intuitive understanding of interpretable word vectors

References

1. Aggarwal, C. C., & Zhai, C.-X. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163–222). Berlin: Springer.
2. Athiwaratkun, B., & Wilson, A. G. (2017). Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1645–1656).
3. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. Preprint. arXiv:1409.0473.
4. Bamler, R., & Mandt, S. (2017). Dynamic word embeddings. Preprint. arXiv:1702.08359.
5. Barkan, O. (2017). Bayesian neural word embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 3135–3143).
6. Barkan, O., & Koenigstein, N. (2016). Item2Vec: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6). Piscataway: IEEE.
7. Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research, 3*, 1137–1155.
8. Bian, W., Li, S., Yang, Z., Chen, G., & Lin, Z. (2017). A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1987–1990). New York: ACM.
9. Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77 (2012).
10. Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06* (pp. 113–120). New York: ACM.
11. Blei, D. M., & Lafferty, J. D. (2009). Topic models. In A. Srivastava & M. Sahami (Eds.), *Text mining: classification, clustering, and applications*. *Data mining and knowledge discovery series*. London: Chapman & Hall.
12. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.
13. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.

14. Bowman, S. R., Angeli, G., Potts, C., & Manning C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg: Association for Computational Linguistics.
15. Boyd-Graber, J., Hu, Y., & Mimno, D. (2017). Applications of topic models. *Foundations and Trends® in Information Retrieval, 11*(2–3), 143–296.
16. Bréal, M. (1897). *Essai de Sémantique: Science des significations*. Paris: Hachette.
17. Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics, 18*(4), 467–479.
18. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. Preprint. arXiv:1409.1259.
19. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, Schwenk, F. H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. Preprint. arXiv:1406.1078.
20. Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-T., Choi, Y., et al. (2018). QuAC: Question answering in context. Preprint. arXiv:1808.07036.
21. Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 160–167). New York: ACM.
22. Cui, H., Sun, R., Li, K., Kan, M.-Y., & Chua, T.-S. (2005). Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 400–407). New York: ACM.
23. Cui, P., Wang, X., Pei, J., & Zhu, W. (2018). A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering, 31*(5), 833–852.
24. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.
25. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint. arXiv:1810.04805.
26. Ding, Y., Liu, Y., Luan, H., & Sun, M. (2017). Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (vol. 1, pp. 1150–1159).
27. Dolan, W. B., & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
28. Dubey, A., Hefny, A., Williamson, S., & Xing, E. P. (2013). A nonparametric mixture model for topic modeling over time. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (pp. 530–538).
29. Dubossarsky, H., Weinshall, D., & Grossman, E. (2017). Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (vol. 1, pp. 1136–1145). Stroudsburg: Association for Computational Linguistics.
30. Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. Preprint. arXiv:1411.4166.
31. Fellbaum, C. (2000). WordNet: An electronic lexical database. *Language, 76*(3), 706.
32. Feng, M., Xiang, B., Glass, M. R., Wang, L., & Zhou, B. (2015). Applying deep learning to answer selection: A study and an open task. Preprint. arXiv:1508.01585.
33. Firth, J. R. (1957). A synopsis of linguistic theory 1930–55. In *Studies in Linguistic Analysis (special volume of the Philological Society)* (vol. 1952–59, pp. 1–32). Oxford: The Philological Society.
34. Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. Preprint. arXiv:1705.03122.

35. Gittens, A., Achlioptas, D., & Mahoney, M. W. (2017). Skip-gram-zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (vol. 1, pp. 69–76).
36. Goller, C., & Kuchler, A. (1996). Learning task-dependent distributed representations by backpropagation through structure. *Neural Networks, 1*, 347–352.
37. Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. Preprint. arXiv:1605.09096.
38. Harris, Z. S. (1954). Distributional structure. *Word, 10*(2–3), 146–162.
39. He, H., Gimpel, K., & Lin, J. (2015). Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1576–1586).
40. He, H., & Lin, J. (2016). Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 937–948).
41. He, K., Girshick, R., & Dollár, P. (2018). Rethinking ImageNet pre-training. Preprint. arXiv:1811.08883.
42. Heilman, M., & Smith, N. A. (2010). Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 1011–1019). Stroudsburg: Association for Computational Linguistics.
43. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
44. Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99* (pp. 50–57). New York: ACM.
45. Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems* (pp. 2042–2050).
46. Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. Preprint. arXiv:1508.01991.
47. Iyyer, M., Yih, W.-T., & Chang, M.-W. (2017). Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (vol. 1, pp. 1821–1831).
48. Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. Preprint. arXiv:1705.03551.
49. 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: Volume 2, Short Papers* (pp. 427–431). Stroudsburg: Association for Computational Linguistics.
50. Kiela, D., Wang, C., & Cho, K. (2018). Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1466–1477).
51. Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751).
52. Kim, Y. (2014). Convolutional neural networks for sentence classification. Preprint. arXiv:1408.5882.
53. Kočiskỳ, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., et al. (2018). The narrativeQA reading comprehension challenge. *Transactions of the Association of Computational Linguistics, 6*, 317–328.
54. Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

55. Kulkarni, V., Al-Rfou, R., Perozzi, B., & Skiena, S. (2015). Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 625–635).
56. Kutuzov, A., Øvrelid, L., Szymanski, T., & Velldal, E. (2018). Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1384–1397). Stroudsburg: Association for Computational Linguistics.
57. Kwiatkowski, T., Palomaki, J., Rhinehart, O., Collins, M., Parikh, A., Alberti, C., et al. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association of Computational Linguistics* (to appear). https://tomkwiat.users.x20web.corp.google.com/papers/natural-questions/main-1455-kwiatkowski.pdf
58. Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to generate a good word embedding. *IEEE Intelligent Systems, 31*(6), 5–14.
59. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. Preprint. arXiv:1603.01360.
60. Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems* (pp. 2177–2185).
61. Li, J., Hu, R., Liu, X., Tiwari, P., Pandey, H. M., Chen, W., et al. (2019). A distant supervision method based on paradigmatic relations for learning word embeddings. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-019-04071-6
62. Li, Q., Uprety, S., Wang, B., & Song, D. (2018). Quantum-inspired complex word embedding. In *Proceedings of the Third Workshop on Representation Learning for NLP, Melbourne* (pp. 50–57). Stroudsburg: Association for Computational Linguistics.
63. Lin, D., & Wu, X. (2009). Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2* (pp. 1030–1038). Stroudsburg: Association for Computational Linguistics.
64. Lin, R., Liu, S., Yang, M., Li, M., Zhou, M., & Li, S. (2015). Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 899–907).
65. Lipton, Z. C. (2016). The mythos of model interpretability. Preprint. arXiv:1606.03490.
66. Lucy, L., & Gauthier, J. (2017). Are distributional representations ready for the real world? Evaluating word vectors for grounded perceptual meaning. Preprint. arXiv:1705.11168.
67. Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers, 28*(2), 203–208.
68. Melucci, M. (2015). *Introduction to information retrieval and quantum mechanics*. Berlin: Springer.
69. Melucci, M., & van Rijsbergen, C. J. (2011). *Quantum mechanics and information retrieval* (chap. 6, pp. 125–155). Berlin: Springer.
70. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Preprint. arXiv:1301.3781.
71. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111–3119).
72. Mimno, D. (2012). Computational historiography. *Journal on Computing and Cultural Heritage, 5*(1), 1–19.
73. Mitra, B., & Craswell, N. (2017). Neural models for information retrieval. Preprint. arXiv:1705.01509.
74. Mitra, B., & Craswell, N. (2018). An introduction to neural information retrieval.*Foundations and Trends in Information Retrieval, 13*(1), 1–126.
75. Mnih, A., & Hinton, G. (2007). Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning* (pp. 641–648). New York: ACM.

76. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., et al. (2016). MS MARCO: A human generated machine reading comprehension dataset. Preprint. arXiv:1611.09268.
77. Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems* (pp. 6338–6347).
78. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (vol. 14, pp. 1532–1543).
79. Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics* (pp. 183–190). Stroudsburg: Association for Computational Linguistics.
80. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. Preprint. arXiv:1802.05365.
81. Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence, 46*(1–2), 77–105.
82. Punyakanok, V., Roth, D., & Yih, W.-T. (2004). Mapping dependencies trees: An application to question answering. In *Proceedings of AI&Math 2004* (pp. 1–10).
83. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language\understanding_paper.pdf
84. Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. Preprint. arXiv:1806.03822.
85. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. Preprint. arXiv:1606.05250.
86. Rao, J., He, H., & Lin, J. (2016). Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 1913–1916). New York: ACM.
87. Reddy, S., Chen, D., & Manning, C. D. (2018). CoQA: A conversational question answering challenge. Preprint. arXiv:1808.07042.
88. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval, 3*(4), 333–389.
89. Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.
90. Rudolph, M., & Blei, D. (2018). Dynamic Bernoulli embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference* (pp. 1003–1011).
91. Rudolph, M., Ruiz, F., Mandt, S., & Blei, D. (2016). Exponential family embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16* (pp. 478–486). Red Hook: Curran Associates Inc.
92. Saha, A., Pahuja, V., Khapra, M. M., Sankaranarayanan, K., & Chandar, S. (2018). Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. Preprint. arXiv:1801.10314.
93. Sala, F., De Sa, C., Gu, A., & Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning* (pp. 4457–4466).
94. 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* (pp. 298–307).
95. Severyn, A., & Moschitti, A. (2013). Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 458–467).
96. Severyn, A., & Moschitti, A. (2015). Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 373–382). New York: ACM.
97. Shen, G., Yang, Y., & Deng, Z.-H. (2017). Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1179–1189).

98. Shi, B., Lam, W., Jameel, S., Schockaert, S., & Lai, K. P. (2017). Jointly learning word embeddings and latent topics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17* (pp. 375–384). New York: ACM.
99. Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. In *Handbook of latent semantic analysis* (pp. 424–440).
100. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems* (pp. 3104–3112).
101. Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. Preprint. arXiv:1503.00075.
102. Talmor, A., & Berant, J. (2018). The web as a knowledge-base for answering complex questions. Preprint. arXiv:1803.06643.
103. Tay, Y., Luu, A. T., & Hui, S. C. (2017). Enabling efficient question answer retrieval via hyperbolic neural networks. CoRR abs/1707.07847.
104. Tay, Y., Phan, M. C., Tuan, L. A., & Hui, S. C. (2017). Learning to rank question answer pairs with holographic dual LSTM architecture. Preprint. arXiv:1707.06372.
105. Tay, Y., Tuan, L. A., & Hui, S. C. (2018). Multi-cast attention networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2299–2308). New York: ACM.
106. Tran, Q. H., Lai, T., Haffari, G., Zukerman, I., Bui, T., & Bui, H. (2018). The context-dependent additive recurrent neural net. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (vol. 1, pp. 1274–1283).
107. Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., & Suleman, K. (2016). NewsQA: A machine comprehension dataset. Preprint. arXiv:1611.09830.
108. Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., et al. (2017). NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP* (pp. 191–200). Stroudsburg: Association for Computational Linguistics.
109. Upadhyay, S., Chang, K. W., Taddy, M., Kalai, A., & Zou, J. (2017). Beyond bilingual: Multi-sense word embeddings using multilingual context. Preprint. arXiv:1706.08160.
110. Van Rijsbergen, C. J. (2004). *The geometry of information retrieval*. Cambridge: Cambridge University Press.
111. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).
112. Vilnis, L., & McCallum, A. (2014). Word representations via Gaussian embedding. Preprint. arXiv:1412.6623.
113. Vorontsov, K., Potapenko, A., & Plavin, A. (2015). Additive regularization of topic models for topic selection and sparse factorization. In *International Symposium on Statistical Learning and Data Sciences* (pp. 193–202). Berlin: Springer.
114. Wang, B., Li, Q., Melucci, M., & Song, D. (2019). Semantic Hilbert space for text representation learning. Preprint. arXiv:1902.09802.
115. Wang, B., Niu, J., Ma, L., Zhang, Y., Zhang, L., Li, J., et al. (2016). A Chinese question answering approach integrating count-based and embedding-based features. In *Natural Language Understanding and Intelligent Applications* (pp. 934–941). Cham: Springer.
116. Wang, D., & Nyberg, E. (2015). A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (vol. 2, pp. 707–712).
117. Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., & Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. Preprint. arXiv:1805.03616.

118. Wang, M., & Manning, C. D. (2010). Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 1164–1172). Stroudsburg: Association for Computational Linguistics.
119. Wang, M., Smith, N. A., & Mitamura, T. (2007). What is the jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.
120. Wang, Z., Hamza, W., & Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. Preprint. arXiv:1702.03814.
121. Wong, S. K. M., Ziarko, W., & Wong, P. C. N. (1985). Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '85* (pp. 18–25). New York: ACM.
122. Yang, L., Ai, Q., Guo, J., & Croft, W. B. (2016). aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 287–296). New York: ACM.
123. Yang, Y., Yih, W.-T., & Meek, C. (2015). WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2013–2018).
124. Yao, X., Van Durme, B., Callison-Burch, C., & Clark, P. (2013). Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 858–867).
125. Yao, Z., Sun, Y., Ding, W., Rao, N., & Xiong, H. (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18* (pp. 673–681). New York: ACM.
126. Yih, W. T., Chang, M. W., Meek, C., & Pastusiak, A. (2013). Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (vol. 1, pp. 1744–1753).
127. Yin, W., & Schütze, H. (2015). Learning meta-embeddings by using ensembles of embedding sets. Preprint. arXiv:1508.04257.
128. Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). Deep learning for answer sentence selection. Preprint. arXiv:1412.1632.
129. Zamani, H., & Croft, W. B. (2017). Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'17* (pp. 505–514). New York: ACM.
130. Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems, 22*(2), 179–214.
131. Zhang, P., Niu, J., Su, Z., Wang, B., Ma, L., & Song, D. (2018). End-to-end quantum-like language models with application to question answering. In *The Thirty-Second AAAI Conference on Artificial Intelligence*. Menlo Park: Association for the Advancement of Artificial Intelligence.
132. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649–657). Cambridge: MIT Press.