Chapter 4
Deep Learning and Its Applications to Natural Language Processing

Haiqin Yang, Linkai Luo, Lap Pong Chueng, David Ling, and Francis Chin

Abstract Natural language processing (NLP), utilizing computer programs to process large amounts of language data, is a key research area in artificial intelligence and computer science. Deep learning technologies have been well developed and applied in this area. However, the literature still lacks a succinct survey, which would allow readers to get a quick understanding of (1) how the deep learning technologies apply to NLP and (2) what the promising applications are. In this survey, we try to investigate the recent developments of NLP, centered around natural language understanding, to answer these two questions. First, we explore the newly developed word embedding or word representation methods. Then, we describe two powerful learning models, Recurrent Neural Networks and Convolutional Neural Networks. Next, we outline five key NLP applications, including (1) part-of-speech tagging and named entity recognition, two fundamental NLP applications; (2) machine translation and automatic English grammatical error correction, two applications with prominent commercial value; and (3) image description, an application requiring technologies of both computer vision and NLP. Moreover, we present a series of benchmark datasets which would be useful for researchers to evaluate the performance of models in the related applications.

Keywords Deep learning · Natural language processing · Word2Vec · Recurrent neural networks · Convolutional neural networks

H. Yang (✉) · L. Luo · L. P. Chueng · D. Ling · F. Chin
Department of Computing, Deep Learning Research and Application Centre, Hang Seng Management College, Sha Tin, Hong Kong
e-mail: hyang@hsmc.edu.hk; linkailuo@hsmc.edu.hk; lpcheung@hsmc.edu.hk; davidling@hsmc.edu.hk; francischin@hsmc.edu.hk

K. Huang et al. (eds.), *Deep Learning: Fundamentals, Theory and Applications*, Cognitive Computation Trends 2, https://doi.org/10.1007/978-3-030-06073-2_4

4.1 Introduction

Deep learning has revived neural networks and artificial intelligence technologies to effectively learn data representation from the original data (LeCun et al. 2015; Goodfellow et al. 2016). Excellent performance has been reported in speech recognition (Graves et al. 2013) and computer vision (Krizhevsky et al. 2017). Now, much effort has now turned to the area of natural language processing.

Natural language processing (NLP), utilizing computer programs to process large amounts of language data, is a key research area in artificial intelligence and computer science. Challenges of NLP include speech recognition, natural language understanding, and natural language generation. Though much effort has been devoted in this area, the literature still lacks a succinct survey, which would allow readers to get a quick understanding of how the deep learning technologies apply to NLP and what the interesting applications are.

In this survey, we try to investigate recent development of NLP to answer the above two questions. We mainly focus on the topics that tackle the challenge of natural language understanding. We will divide the introduction into the following three aspects:

- summarizing the neural language models to learn word vector representations, including **Word2vec** and **Glove** (Mikolov et al. 2013a,b; Pennington et al. 2014),
- introducing the powerful tools of the recurrent neural networks (RNNs) (Elman 1990; Chung et al. 2014; Hochreiter and Schmidhuber 1997) and the convolutional neural networks (CNNs) (Kim 2014; dos Santos and Gatti 2014; Gehring et al. 2017), for language models to capture dependencies in languages. More specifically, we will introduce two popular extensions of RNNs, i.e., the long short-term memory (LSMT) (Hochreiter and Schmidhuber 1997) network and the Gated Recurrent Unit (GRU) (Chung et al. 2014) network, and briefly discuss the efficiency of CNNs for NLP.
- outlining and sketching the development of five key NLP applications, including part-of-speech (POS) tagging (Collobert et al. 2011; Toutanova et al. 2003), named entity recognition (NER) (Collobert et al. 2011; Florian et al. 2003), machine translation (Bahdanau et al. 2014; Sutskever et al. 2014), automatic English grammatical error correction (Bhirud et al. 2017; Hoang et al. 2016; Manchanda et al. 2016; Ng et al. 2014), and image description (Bernardi et al. 2016; Hodosh et al. 2013; Karpathy and Fei-Fei 2017).

Finally, we present a series of benchmark datasets which are popularly applied in the above models and applications, while concluding the whole article with some discussions. We hope this short review of the recent progress of NLP can help researchers new to the area to quickly enter this field.

4.2 Learning Word Representations

A critical issue of NLP is to effectively represent the features from the original text data. Traditionally, the numerical statistics, such as term frequency or term frequency inverse document frequency (tf-idf), are utilized to determine the importance of a word. However, in NLP, the goal is to extract the semantic meaning from the given corpus. In the following, we will introduce the state-of-the-art word embedding methods, including **word2vec** (Mikolov et al. 2013a) and **Glove** (Pennington et al. 2014).

Word embeddings (or word representations) are arguably the most widely known technique in the recent history of NLP. Formally, a word embedding or a word representation is represented as a vector of real numbers for each word in the vocabulary. There are various approaches to learn word embeddings, which force similar words to be as close as possible in the semantic space. Among them **word2vec** and **Glove** have attracted a great amount of attention in recent 4 years. These two methods are based on the distributional hypothesis (Harris 1954), where words appearing in similar contexts tend to have similar meaning, and the concept that one can know a word by the company it keeps (Firth 1957).

Word2vec (Mikolov et al. 2013a) is not a new concept; however, it gained popularity only after two important papers Mikolov et al. (2013a,b) were published in 2013. **Word2vec** models are constructed by shallow (only two-layer) feedforward neural networks to reconstruct linguistic contexts of words. The networks are fed a large corpus of text and then produce a vector space that is shown to carry the semantic meanings. In Mikolov et al. (2013a), two **wor2vec** models, i.e., Continuous Bag of Words (CBOW) and skip-gram, are introduced. In CBOW, the word embeddings is constructed through a supervised deep learning approach by considering the fake learning task of predicting a word by its surrounding context, which is usually restricted to a small window of words. In skip-gram, the model utilizes the current word to predict its surrounding context words. Both approaches take the value of the vector of a fixed-size inner layer as the embedding. Note that the order of context words does not influence the prediction in both settings. According to Mikolov et al. (2013a), CBOW trains faster than skip-gram, but skip-gram does better job in detecting infrequent words.

One main issue of **word2vec** is the high computational cost due to the huge amount of corpora. In Mikolov et al. (2013b), hierarchical softmax and negative sampling are proposed to address the computational issue. Moreover, to enhance computational efficiency, several tricks are adopted: including (1) eliminating most frequent words such as “a”, “the”, and etc., as they provide less informational value than rare words; and (2) learning common phrases and treating them as single words, e.g., “‘New York” is replaced by “New_York”. More details about the algorithms and the tricks can be found in Rong (2014).

An implementation of **word2vec** in C language is available in the Google Code Archive[1] and its Python version can be downloaded in **gensim**.[2]

Glove (Pennington et al. 2014) is based on the hyperthesis that related words often appear in the same documents and looks at the ratio of the co-occurrence probability of two words rather than their co-occurrence probability. That is, the **Glove** algorithm involves collecting word co-occurrence statistics in the form of a word co-occurrence matrix X, whose element X_{ij} represents how often word i appears in the context of word j. It then defines a weighted cost function to yield the final word vectors for all the words in the vocabulary. The corresponding source code for the model and pre-trained word vectors are available here.[3]

Word embeddings are widely adopted in a variant of NLP tasks. In Kim (2014), the pre-trained **word2vec** is directly employed for sentence-level classifications. In Hu et al. (2017, 2018), the pre-trained **word2vec** is tested in predicting the quality of online health expert question-answering services. It is noted that the determination of word vector dimensions is mostly task-dependent. For example, a smaller dimensionality works better for more syntactic tasks such as named entity recognition (Melamud et al. 2016) or part-of-speech (POS) tagging (Plank et al. 2016), while a larger dimensionality is more effective for more semantic tasks such as sentiment analysis (Ruder et al. 2016).

4.3 Learning Models

A long-running challenge of NLP models is to capture dependencies, especially the long-distance dependencies, of sentences. A natural idea is to apply the powerful sequence data learning models, i.e., the recurrent neural networks (RNNs) (Elman 1990), in language models. Hence, in the following, we will introduce RNNs and more especially, the famous long short-term memory (LSMT) network (Hochreiter and Schmidhuber 1997) and the recently proposed Gated Recurrent Unit (GRU) (Chung et al. 2014). Moreover, we will briefly describe convolutional neural networks (CNNs) in NLP, which can be efficiently trained.

4.3.1 Recurrent Neural Networks (RNNs)

RNNs are powerful tools for language models, since they have the ability to capture long-distance dependencies in sequence data. The idea to model long-distance dependencies is quite straightforward, that is, to simply use the previous hidden

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

[2] https://radimrehurek.com/gensim/

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

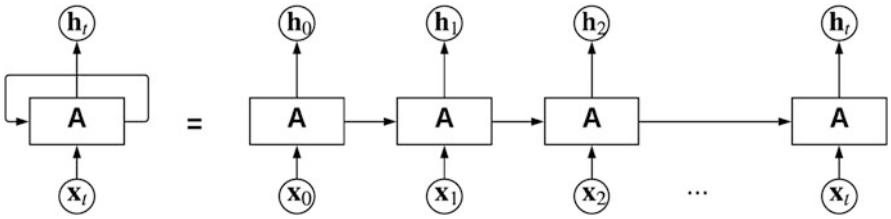

Fig. 4.1 Architecture of RNN

state $\mathbf{h}_{t-1}$ as input when calculating the current hidden state $\mathbf{h}_t$. See Fig. 4.1 for an illustration, where the recursive node can be unfolded into a sequence of nodes.

Mathematically, an RNN can be defined by the following equation:

$$\mathbf{h}_t = \begin{cases} \tanh\left(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h\right) & t \geq 1, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{4.1}$$

where $\mathbf{x}_t$ is the t-th sequence input, $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ is the bias vector. At the t-th (≥ 1) time stamp, the only difference between an RNN and a standard neural network lies in the additional connection $\mathbf{W}_{hh}\mathbf{h}_{t-1}$ from the hidden state at time step $t-1$ to that at the t time stamp.

Though RNNs are simply and easy to compute, they encounter the vanishing gradient problem, which results in little change in the weights and thus no training, or the exploding gradient problems, which results in large changes in the weights and thus unstable training. These problems typically arises in the back propagation algorithm for updating the weights of the networks (Pascanu et al. 2013). In Pascanu et al. (2013), a gradient norm clipping strategy is proposed to deal with exploding gradients and a soft constraint is proposed for the vanishing gradients problem. The proposed method does not utilize the information in a whole.

RNNs are very effective for sequence processing, especially for short-term dependencies, i.e., neighboring contexts. However, if the sequence is long, the long term information is lost. One successful and popular model is to modify the RNN architecture, producing namely the long short-term memory (LSMT) (Hochreiter and Schmidhuber 1997) network. The creativity of LSTM is to introduce the memory cell $\boldsymbol{c}$ and gates that controlling the signal flows in the architecture. See the illustrated architecture in Fig. 4.2a and the corresponding formulas as follows:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{4.2}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i\right) \tag{4.3}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{4.4}$$

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right) \tag{4.5}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{4.6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{4.7}$$

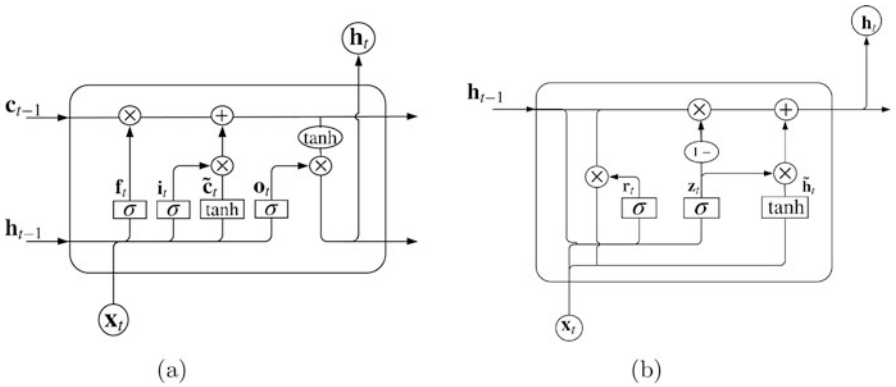

Fig. 4.2 Architecture of (**a**) LSTM and (**b**) GRU

Equations (4.2), (4.3) and (4.4) correspond to the forget gate, the input gate, and the output gate, respectively. σ is the logistic function outputting the value in the range [0, 1], $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector, respectively, and $\odot$ is the element wise multiplication operator. Equations 4.2, 4.3 and 4.4 corresponds to the forget gate, input gate and output gate, respectively. The function of these gates, as their name indicate, is either allow all signal information to pass through (the gate output equals 1) or block it from passing (the gate output equals 0).

In addition to the standard LSTM model described above, a few LSTM variants have been proposed and proven to be effective. Among them, the Gated Recurrent Unit (GRU) (Chung et al. 2014) network is one of the most popular ones. GRU is simpler than a standard LSTM as it combines the input gate and the forget gate into a single update gate. See the illustrated architecture in Fig. 4.2b and the corresponding formulas as follows:

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r\right) \tag{4.8}$$

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z\right) \tag{4.9}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\right) \tag{4.10}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t. \tag{4.11}$$

Compared to the LSTM, the GRU has slightly fewer parameters and also does not have a separate "cell" to store intermediate information. Due to its simplicity, GRU has been extensively used in many sequence learning tasks to conserve memory or computation time. Besides GRU, there are a few variants that share similar but slightly different architecture as LSTM. More details can be found in Gers and Schmidhuber (2000), Koutník et al. (2014), Graves et al. (2017), and Józefowicz et al. (2015).

4.3.2 Convolutional Neural Networks (CNNs)

While RNNs are the ideal choices for many NLP tasks, they have an inherent limitation. Most RNNs rely on bi-directional encoders to build representations of both past and future contexts (Bahdanau et al. 2014; Zhou et al. 2016). They can only process one word at a time. It is less natural to utilize the parallelization architecture of GPU computation in the training and the hierarchical representations over the input sequence (Gehring et al. 2017). To tackle these challenges, researchers have proposed the convolutional architecture for neural machine translation (Gehring et al. 2017). The work borrows the idea of CNNs which utilize layers with convolving filters to extract local features and have been successfully applied in image processing (LeCun et al. 1998). In the convolutional architecture, the input elements $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ are embedded in a distributional space as $\mathbf{w} = (w_1, w_2, \ldots, w_m))$, where $w_j \in \mathbb{R}^f$. The final input element representation is computed by $\mathbf{e} = (w_1 + p_1, w_2 + p_2, \ldots, w_m + p_m)$, where $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ is the embedded representation of the absolute position of input elements with $p_j \in \mathbb{R}^f$. A convolutional block structure is applied in the input elements to output the decoder network $\mathbf{g} = (g_1, g_2, \ldots, g_n)$. The proposed architecture is reported to outperform the previous best result by 1.9 BLEU on WMT'16 English-Romanian translation (Zhou et al. 2016).

CNNs not only can compute all words simultaneously by taking advantage of GPU parallelization computation, which shows much faster training than RNNs, but they also show better performance than the LSTM models (Zhou et al. 2016). Other NLP tasks, such as sentence-level sentiment analysis (Kim 2014; dos Santos and Gatti 2014), character-level machine translation (Costa-Jussà and Fonollosa 2016), and simple question answering (Yin et al. 2016), also demonstrate the effectiveness of CNNs.

4.4 Applications

In the following, we present the development of five key NLP applications: part-of-speech (POS) tagging and named entity recognition (NER) are two fundamental NLP applications, which can enrich the analysis of other NLP applications (Collobert et al. 2011; Florian et al. 2003; Toutanova et al. 2003); machine translation and automatic English grammatical error correction are two applications containing direct commercial value (Bahdanau et al. 2014; Bhirud et al. 2017; Hoang et al. 2016; Manchanda et al. 2016; Ng et al. 2014; Sutskever et al. 2014); and image description, an attractive and significant application requiring the techniques of both computer vision and NLP (Bernardi et al. 2016; Hodosh et al. 2013; Karpathy and Fei-Fei 2017).

4.4.1 Part-of-Speech (POS) Tagging

Part-of-speech (POS) tagging (Collobert et al. 2011) aims at labeling (assocating) each word with a unique tag that indicates its *syntactic role*, e.g., plural noun, adverbs, etc. The POS tags are usually utilized as common input features for various NLP tasks, e.g., information retrieval, machine translation (Ueffing and Ney 2003), grammar checking (Ng et al. 2014), etc.

Nowadays, the most common used POS category is the tag set in the Penn Treebank Project, which defines 48 different tags (Marcus et al. 1993). They are commonly used in various NLP libraries, such as NLTK[4] in Python, Stanford tagger,[5] and Apache OpenNLP.[6]

The existing algorithms for tagging can be generally categorized into two groups, the rule-based group and the stochastic group. The rule-based methods such as the Eric Brills tagger (Brill 1992) and the disambiguation rules in LanguageTool,[7] are usually hand-crafted, derived from corpus, or developed collaboratively (e.g., for LanguageTool). The rule-based methods can achieve a pretty low error rate (Brill 1992), but generally, they are still less sophisticated when compared with stochastic taggers. In contrast, stochastic taggers, such as the Hidden Markov Model (HMM) (Brants 2000) and the Maximum Entropy Markov Model (MEMM) (McCallum et al. 2000), model the sequence of POS tags as the hidden states, which can be learned from the observed word sequence of sentences. The probability of co-occurrence of words and tags is modeled by HMM (Brants 2000) and the conditional probability of tags given the words is modeled by MEMM (McCallum et al. 2000) to output the corresponding tags.

Later, more advanced methods have been proposed to improve both HMM and MEMM. The methods include utilizing bidirectional cyclic dependency network tagger (Manning 2011) and using other linguistic features (Jurafsky and Martin 2017). More than 96% accuracy was reported by both HMM (Brants 2000) and MEMM (Manning 2011). More state-of-the-art performances can be found on internet.[8]

4.4.2 Named Entity Recognition (NER)

Named entity recognition (NER) is a classic NLP task that seeks to locate and classify named entities such as person names, organizations, locations, numbers,

[4] http://www.nltk.org/

[5] https://nlp.stanford.edu/software/tagger.shtml

[6] https://opennlp.apache.org/

[7] http://wiki.languagetool.org/developing-a-disambiguator

[8] https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

dates, etc. from the text corpora. Most existing NER taggers are built on linear statistical models, such as Hidden Markov Models (McCallum et al. 2000) and Conditional Random Field (Lafferty et al. 2001). Traditional NER techniques heavily rely on hand-crafted features for the taggers and only apply for small corpora (Chieu and Ng 2002).

Nowadays, due to the development of deep learning technologies, a variety of neural network models, such as LSTM and CNN, have been proposed to establish the tagger models (Huang et al. 2015; Lample et al. 2016). Unlike the standard neural networks for conventional classification whose final layer is a softmax, the NN based named entity models utilize a linear-chain CRF to model the dependencies across the word sequence for NER. In Huang et al. (2015) and Lample et al. (2016), the sequence tagging model consists of a bidirectional LSTM network and a CRF layer (BI-LSTM-CRF). In Ma and Hovy (2016), the BI-LSTM-CRF is modified by adding a character-based CNNs at the bottom of BI-LSTM. The CNNs are used to encode the characters of a word into its character-level representation. The added character-level information, together with word-level representation is then fed into the bidirectional LSTM. This so-called Bi-directional LSTM-CNNs-CRF architecture is reported to be better than the BI-LSTM-CRF one. Similar publications have been generated to implement the LSTM network and the CRF layer for NER tasks (Chiu and Nichols 2016; Yang et al. 2016; Wang et al. 2015).

4.4.3 Neural Machine Translation

The objective of machine translation (MT) is to translate text or speech from one language to another one. Conventional MT utilizes statistical models whose parameters are inferred from bilingual text corpora. Recently, a major development in MT is the adoption of sequence to sequence learning models, promoting the state-of-art technique called neural machine translation (NMT) (Wu et al. 2016; Gehring et al. 2017; Vaswani et al. 2017). NMT has been proven great success owing to the rapid development of deep learning technologies, whose architecture is comprised of an encoder-decoder model (Sutskever et al. 2014), and an attention mechanism (Bahdanau et al. 2014).

An encoder model RNN_{enc} provides a representation of the source sentence by inputing a sequence of source words $\mathbf{x} = (x_1, \ldots, x_m)$ and producing a sequence of hidden states $\mathbf{h} = (h_1, \ldots, h_m)$. According to Sutskever et al. (2014), a bidirectional RNN_{enc} is usually favored to reduce long sentence dependencies, and the final state $\mathbf{h}$ is the concatenation of the states produced by forward and backward RNNs, $\mathbf{h} = \left[\overrightarrow{\mathbf{h}}; \overleftarrow{\mathbf{h}}\right]$. The decoder is also a recurrent neural network, RNN_{dec}, which predicts the probability of a target word of a sentence y_k, based on the hidden state $\mathbf{h}$, the previous words $\mathbf{y}_{<k} = (y_1, \ldots, y_{k-1})$, the recurrent hidden state in the decoder RNN $\boldsymbol{s}_k$, and the context vector $\mathbf{c}_k$. The context vector $\mathbf{c}_k$ is also called the attention vector, which is computed as a weighted vector of the source hidden state

h: $\sum_{j=1}^{m} \alpha_{ij} h_j$, where m is the length of source sentence, and α_{ij} is the attention weight. The attention weight can be calculated in the fashion of concatenation of bi-directional encoder (Bahdanau et al. 2014) or a simpler version with a location-based function on the target hidden state (Luong et al. 2015b). Finally, the decoder outputs a distribution over a fixed-size vocabulary through softmax approximation:

$$P(y_k|\mathbf{y}_{<k}, \mathbf{x}) = \text{softmax}\left(g(y_{k-1}, \mathbf{c}_k, \mathbf{s}_k)\right) \tag{4.12}$$

where g is a non-linear function. The encoder-decoder and attention-driven model is trained end-to-end by optimizing the negative log likelihood of the target words using stochastic gradient descent (SGD).

The tuning of hyper-parameters of NMT model is crucial to the performance of translation. In Britz et al. (2017), it is concluded that a higher dimensional embedding such as 2,048 usually yields the best performance. Nevertheless, small dimensionality such as 128 shall surprisingly perform well and converge much faster for some tasks. The depth of encoder and decoder is not necessarily deeper than four layers, although in Wu et al. (2016), eight layers are employed. Bidirectional encoders always outperform unidirectional ones as they are able to create representations that take both past and future sequence words into account. The comparison in Wu et al. (2016) also shows that LSTM cells consistently beat GRU cells. Moreover, beam search (Wiseman and Rush 2016) is commonly used in most NMT tasks to output more precise target words. Usually, the well-tuned beam search size ranges from 5 to 10. The algorithm optimizer in the training will also affect the performance. Adam (Kingma and Ba 2014) optimizer with a fixed learning rate (smaller than 0.01) without decay seems effective and shows fast convergence. In some tasks, however, standard SGD with scheduling will generally lead to better performance although the convergence is relatively slow (Ruder 2016). There are other hyper-parameters that directly relate to the model performance, to name a few, dropout (Srivastava et al. 2014), layer normalization (Ba et al. 2016), residual connection of layers (He et al. 2016), etc.

Next, we summarize some aspects in advancing NMT. The first issue is to restrict the size of the vocabulary. Though NMT is an open vocabulary problem, the number of target words of NMT must be limited, because the complexity of training an NMT model increases as the number of target words increases. In practice, the target vocabulary size K is often in the range of $30k$ (Bahdanau et al. 2014) to $80k$ (Sutskever et al. 2014). Any word out of the vocabulary is represented as an *unknown* word, denoted by *unk*. The traditional NMT model works well if there are fewer unknown words in the target sentences, but it has been observed that the performance of translation degrades dramatically if there are too many unknown words (Jean et al. 2015). An intuitive solution to address this problem is to use a larger vocabulary, while simultaneously reducing the computational complexity using sampling approximations (Jean et al. 2015; Mi et al. 2016; Ji et al. 2015). Other researcher reported that the unknown word problem can be addressed alternatively without expanding vocabulary. For example, one can replace the unknown word with special token *unk*, and then post-process the target sentence

by copying the *unk* from source sentence or applying word translation to the unknown word (Luong et al. 2015c). Instead of implementing word-based neural machine translation, other researchers proposed to using character-based NMT to eliminate unknown words (Costa-Jussà and Fonollosa 2016; Chung et al. 2016), or using a hybrid method – a combination of word-level and character-level NMT model (Luong and Manning 2016). The implementation of subword units also shows significant effectiveness in reducing the vocabulary size (Sennrich et al. 2016b). The algorithm, called byte pair encoding (BPE), starts with a vocabulary of characters, and replaces the most frequent n-gram pairs with a new n-gram.[9] To summarize, the word-level, BPE-level and character-level vocabulary forms the fundamental treatment of neural machine translation practice.

The second issue is about the training corpus. As widely noted, one of the major factors behind the success of NMT is the availability of high quality parallel corpora. How to include more other data sources into NMT training has become critical and drawn great attention recently. Inspired by statistical machine translation, the researchers improve the translation quality by leveraging abundant monolingual corpora for neural machine translation (Gucehre et al. 2015; Sennrich et al. 2016a). Two recent publications propose an unsupervised machine translation method to utilize monolingual data (Artetxe et al. 2017; Lample et al. 2017). Both methods train a neural machine translation model without any parallel corpora with fairly high accuracy, and establish the future direction for NMT. In Luong et al. (2015a), Johnson et al. (2017), and Firat et al. (2017), the authors use a single NMT model to translate between multiple languages, such that the encoder, decoder and attention modules can be shard across all languages.

The third issue is the implementation of neural machine translation. To deploy neural machine translation systems, one needs to build the encoder-decoder model (with attention mechanism) and to train the end-to-end model on GPUs. Nowadasy, there are quite many toolkits publicly available for research, development and deployment:

- dl4mt-tutorial (based on Theano): https://github.com/nyu-dl/dl4mt-tutorial
- Seq2seq (based on Tensorflow): https://github.com/google/seq2seq
- OpenNMT (based on Torch/PyTorch): http://opennmt.net
- xnmt (based on DyNet): https://github.com/neulab/xnmt
- Sockeye (based on MXNet): https://github.com/awslabs/sockeye
- Marian (based on C++): https://github.com/marian-nmt/marian
- nmt-keras (based on Keras): https://github.com/lvapeab/nmt-keras

[9]The source code can be found at https://github.com/rsennrich/nematus.

4.4.4 Automatic English Grammatical Error Correction

Since English is not the first language of many people in the world, to facilitate the writing, grammar checkers have been developed. Some commercial or freeware such as Microsoft Word, Grammarly,[10] LanguageTool,[11] Apache Wave,[12] and Ginger,[13] can provide grammar checking services. However, due to various exceptions and rules in natural languages, these grammar checkers are still fall far short of human English teachers.

To boost the development of grammatical error checking and correction, various shared tasks and focused sessions were launched to attract researchers' interests and contributions. The tasks include the Helping Our Own (HOO) Shared Task in 2011 (Dale and Kilgarriff 2011), the CoNLL Shared Task in 2013 (Ng et al. 2013) and 2014 (Ng et al. 2014), respectively, and the AESW Shared Task in 2016 (Daudaravicius et al. 2016). Each of the shared tasks provided the original text corpus and the corresponding ones corrected by human editors. The dataset of CoNLL Shared Task 2013 and 2014 is a collection of 1,414 marked student essays from the National University of Singapore, where all the students are non-native English speakers. The detected grammatical errors are classified into 28 types. Meanwhile, the datasets of the HOO and the AESW shared tasks are extracted from published papers and proceedings of conferences. The HOO task is a collection of fractional texts from 19 published papers, while the AESW one is a collection of shuffled sentences generated from 9,919 published papers (mainly from physics and mathematics).

Recently, various methods have been proposed to correct the grammatical errors (Manchanda et al. 2016; Rozovskaya and Roth 2016; Bhirud et al. 2017; Ng et al. 2014), which can be categorized into three main types: (1) the rule-based approach, (2) the statistical approach, (3) the machine translation approach. The rule-based approach utilizes rules in the detection of mistakes. The rules are usually hand-crafted rules, inputted manually based on different cases. Most of them use pattern matching, dependency parse tree, as well as POS to find the grammatical errors, e.g., subject-verb-agreement. The rule-based approach can be found in LanguageTool (Daniel 2003) and several systems in the CoNLL shared task (Ng et al. 2014). It is usually too time-consuming to generating the hand-crafted rules. Hence, researchers turn to the statistical approaches, which can learn the rules from large corpora such as the English Wikipedia dump, the Google Book N-gram, Web1T corpus, Cambridge Learner Corpus, and English Giga Word corpus, et. Some typical methods include (1) extracted tri-grams with low frequency and particular patterns from the Web1T corpus (Wu et al. 2013), (2) utilizing the

[10] https://www.grammarly.com/

[11] https://languagetool.org/

[12] https://incubator.apache.org/wave/

[13] Ginger

three tokens around the target article and the Averaged Perceptron to suggest the correct article (Rozovskaya and Roth 2010), and (3) detecting grammar errors by comparing non-existent bi-grams in Google Book n-gram corpus (Nazar and Renau 2012). The statistical approaches are often favorable in grammar checking because they only require a big corpus from native English users. In contrast, the machine translation approaches need a big parallel corpus to extract the corresponding rules. Due to the development of deep learning technologies, the machine translation approaches become prevalent in correcting the grammatical errors. These methods utilize the methods mentioned in Sect. 4.4.3 to feed the problematic sentences and output the correct ones. For example, the AMU team (Ng et al. 2014) utilized the Phrase-based machine translation in the detection for the task in CoNLL 2014 while CNN with LSTM is applied to tackle the correction problem (Schmaltz et al. 2016). In Schmaltz et al. (2016), the input and output sentences are encoded by some additional tags to fit the requirement of NMT. For example, the input sentence "The models works <eos>" corresponds to the output sentence "The models ~~works~~ <ins>work</ins> <eos>", where <del>, <ins>, and <eos> are the tags denoting the deletion operation, the insertion operation, and the end of sentence. More recent proposals for machine translation methods and some fair comparisons can be referred to Junczys-Dowmunt and Grundkiewicz (2016), Hoang et al. (2016), and Rozovskaya and Roth (2016).

4.4.5 Image Description

Image description (Karpathy and Fei-Fei 2017; Vinyals et al. 2017; Xu et al. 2015) is a challenging and active research topic which requires techniques from both computer vision and natural language processing. Its goal is to automatically generate natural language descriptions of images on the corresponding regions. Researchers have proposed different models to learn about the correspondences between language and visual data. For example, in Karpathy and Fei-Fei (2017), a multimodal RNN architecture is proposed to align a modality trained by CNNs over image regions with a modality trained by bidirectional RNNs over sentences. In Vinyals et al. (2017), CNNs are applied to learn the representation of images while LSTMs are utilized to output the sentences. A direct model is built to maximize the likelihood of the sentence given the image. In Xu et al. (2015), similar to Vinyals et al. (2017), CNNs are applied to generate the representation of images and LSTMs are utilized to produce the captions. The key improvement is to include attention-based mechanisms to further improve the model performance. The performance of image captioning is increased as new methods have been proposed. More details can be referred to Bernardi et al. (2016).

4.5 Datasets for Natural Language Processing

Many datasets have been published in different research domains for natural language processing. We try to provide the basic ones mentioned in previous sections.

4.5.1 Word Embedding

- **word2vec**[14]: The link not only provides the pre-trained vectors in 300-dimensions of 3 million words and phrases, which are trained on Google News dataset (about 100 billion words), but also provide various online available datasets, such as the first billion characters from wikipedia, the latest Wikipedia dump, the WMT11 site, and etc.
- **Glove**[15]: The word vectors are trained by Glove (Pennington et al. 2014). The dataset contains pre-trained vectors trained from sources including Wikipedia, Twitter and some common crawled data.

4.5.2 N-Gram

- **Google Book N-gram**[16]: The dataset contains 1–5-gram counting from printed books in different languages, e.g., English, Chinese, French, Hebre, Italian, etc. Specialized corpora are available for English, like American English, British English, English Fiction, and English One Million. The n-grams are tagged with Part-Of-Speech, and are counted yearly.
- **Web 1T 5-gram**[17]: The dataset, contributed by Google, consists of 1–5-gram counting from accessible websites and yields about 1 trillion tokens. The compressed file size (gzip'ed) is approximately 24 GB.

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

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

[16] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

[17] https://catalog.ldc.upenn.edu/ldc2006t13

4.5.3 Text Classification

- **Reuters Corpora (RCV1, RCV2, TRC2)**[18]: The dataset contains a large collection of Reuters News stories, which is written in five languages and the corresponding translations in six categories. Detailed description can be found in Lewis et al. (2004)
- **IMDB Movie Review Sentiment Classification**[19]: The dataset, consisting of review comments of 50,000 movies, is first tested in Maas et al. (2011) for binary sentiment classification.
- **News Group Movie Review Sentiment Classification**[20]: The datasets were introduced in Pang et al. (2002) and Pang and Lee (2004, 2005) for sentimental analysis. They consist of movie-review documents labeled with respect to their overall sentiment polarity (position or negative) or subjective rating (e.g., "two and a half stars") and sentences labeled with respect to their subjectivity status (subjective or objective) or polarity.

4.5.4 Part-Of-Speech (POS) Tagging

- **Penn Treebank**[21]: The dataset selected 2,499 stories from a three year Wall Street Journal (WSJ) collection of 98,732 stories for syntactic annotation (Marcus et al. 1999).
- **Universal Dependencies**[22]: Universal Dependencies is a project that seeks to develop cross-linguistically consistent treebank annotation for multiple languages. The latest version contains 102 treebanks in 60 languages (Nivre et al. 2017).

4.5.5 Machine Translation

- **Europarl**[23]: The Europarl parallel corpus contains sentences pairs in 21 European languages. Detailed description can be found in Koehn (2005).

[18] http://trec.nist.gov/data/reuters/reuters.html
[19] http://ai.stanford.edu/~amaas/data/sentiment/
[20] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[21] https://catalog.ldc.upenn.edu/ldc99t42
[22] https://catalog.ldc.upenn.edu/LDC2000T43
[23] http://www.statmt.org/europarl/

– **United Nations Parallel Corpus**[24]: The corpus is generated from the official records and other parliamentary documents of the United Nations. These documents are mostly available in the six official languages of the United Nations (Ziemski et al. 2016).

4.5.6 Automatic Grammatical Error Correction

– **NUS Corpus of Learner English (NUCLE)**[25]: The corpus consists of about 1,400 essays written by students at the National University of Singapore. The essays are completely annotated with error tags and corrections by English instructors.
– **AESW 2016 Data Set**[26]: The dataset is a collection of random ordered sentences extracted from 9,919 published journal articles (mainly from physics and mathematics).The sentences are annotated with the changes made by journal editors.

4.5.7 Image Description

– **Flickr8K**[27]: The dataset is standard benchmark for sentence-based image description, consisting of around 8K images crawled from the Flickr.com website, where each image is paired with five different captions to provide clear descriptions of the salient entities and events (Hodosh et al. 2013).
– **Flickr30K**[28]: The dataset is an extended version of Flickr8K and consists of around 30K images while each image containing five descriptions (Plummer et al. 2017).
– **MSCOCO**[29]: The dataset consists of 123,287 images with five different descriptions per image (Lin et al. 2014). Images in the dataset are annotated for 80 categories and provided the bounding boxes around all instances in one of the categories.

[24] https://conferences.unite.un.org/uncorpus
[25] http://www.comp.nus.edu.sg/~nlp/conll14st.html
[26] http://textmining.lt/aesw/index.html
[27] http://nlp.cs.illinois.edu/HockenmaierGroup/8k-pictures.html
[28] http://web.engr.illinois.edu/~bplumme2/Flickr30kEntities/
[29] http://cocodataset.org/

4.6 Conclusions and Discussions

In this survey, we have provided a succinct review of the recent development of NLP, including word representation, learning models, and key applications. Nowadays, **Word2vect** and **Glove** are two main successful methods to learn the word representation in the semantic space. RNNs and CNNs are two mainstreams of learning models to train the NLP models. After exploring the five key applications, we envision the following interesting research topics. First, it is effective to include additional features or results (e.g., POS tagging and NER) to improve the performance for other applications, such as machine translations and automatic grammar correction. Second, it is worth investigating the end-to-end model, which may further improve the model performance. For example, nowadays, the embedded word representation is learned independently to the applications. One may explore new representations which fit for the later applications, e.g., sentimental analysis, text matching. Third, it is promising to explore the advancement of multidisciplinary approaches. For example, in the image description application, one needs the technologies from both computer vision and natural language processing. It is significant to understand both areas and make the breakthrough.

Acknowledgements The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/IDS14/16).

References

Artetxe M, Labaka G, Agirre E, Cho K (2017) Unsupervised neural machine translation. *CoRR*, abs/1710.11041

Ba JL, Kiros R, Hinton EG (2016) Layer normalization. *CoRR*, abs/1607.06450

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473

Bernardi R, Çakici R, Elliott D, Erdem A, Erdem E, Ikizler-Cinbis N, Keller F, Muscat A, Plank B (2016) Automatic description generation from images: a survey of models, datasets, and evaluation measures. J Artif Intell Res 55:409–442

Bhirud SN, Bhavsar R, Pawar B (2017) Grammar checkers for natural languages:a review. Int J Natural Lang Comput 6(4):1

Brants T (2000) Tnt: a statistical part-of-speech tagger. In: ANLC'00, Stroudsburg. Association for Computational Linguistics, pp 224–231

Brill E (1992) A simple rule-based part of speech tagger. In: ANLC, Stroudsburg, pp 152–155

Britz D, Goldie A, Luong M, Le VQ (2017) Massive exploration of neural machine translation architectures. *CoRR*, abs/1703.03906

Chieu LH, Ng TH (2002) Named entity recognition: a maximum entropy approach using global information. In: COLING, Taipei

Chiu JPC, Nichols E (2016) Named entity recognition with bidirectional LSTM-CNNs. TACL 4:357–370

Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555

Chung J, Cho K, Bengio Y (2016) A character-level decoder without explicit segmentation for neural machine translation. In: ACL, Berlin

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa PP (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537

Costa-Jussà MR, Fonollosa JAR (2016) Character-based neural machine translation. In: ACL, Berlin

Dale R, Kilgarriff A (2011) Helping our own: the HOO 2011 pilot shared task. In: ENLG, Nancy, pp 242–249

Daniel N (2003) A rule-based style and grammar checker. Master's thesis, Bielefeld University, Bielefeld

Daudaravicius V, Banchs ER, Volodina E, Napoles C (2016) A report on the automatic evaluation of scientific writing shared task. In: Proceedings of the 11th workshop on innovative use of NLP for building educational applications, BEA@NAACL-HLT 2016, San Diego, 16 June 2016, pp 53–62

dos Santos CN, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: COLING, Dublin, pp 69–78

Elman LJ (1990) Finding structure in time. Cogn Sci 14(2):179–211

Firat O, Cho K, Sankaran B, Yarman-Vural FT, Bengio Y (2017) Multi-way, multilingual neural machine translation. Comput Speech Lang 45:236–252

Firth RJ (1957) A synopsis of linguistic theory 1930–1955. Studies in linguistic analysis. Blackwell, Oxford, pp 1–32

Florian R, Ittycheriah A, Jing H, Zhang T (2003) Named entity recognition through classifier combination. In: Proceedings of the seventh conference on natural language learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, 31 May–1 June 2003, pp 168–171

Gehring J, Auli M, Grangier D, Dauphin Y (2017) A convolutional encoder model for neural machine translation. In: ACL, Vancouver, pp 123–135

Gehring J, Auli M, Grangier D, Yarats D, Dauphin NY (2017) Convolutional sequence to sequence learning. In: ICML, Sydney, pp 1243–1252

Gers AF, Schmidhuber J (2000) Recurrent nets that time and count. In: IJCNN (3), Como, pp 189–194

Goodfellow JI, Bengio Y, Courville CA (2016) Deep learning. Adaptive computation and machine learning. MIT Press, Cambridge

Graves A, Mohamed A, Hinton EG (2013) Speech recognition with deep recurrent neural networks. In: IEEE ICASSP, British Columbia, pp 6645–6649

Greff K, Srivastava KR, Koutník J, Steunebrink RB, Schmidhuber J (2017) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222–2232

Gucehre C, Firat O, Xu K, Cho K, Barrault L, Lin H, Bougares F, Schwenk H, Bengio Y (2015) On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535

Harris Z (1954) Distributional structure. Word 10(23):146–162

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR, Las Vegas, pp 770–778

Hoang TD, Chollampatt S, Ng TH (2016) Exploiting n-best hypotheses to improve an SMT approach to grammatical error correction. In: IJCAI, pp 2803–2809

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hodosh M, Young P, Hockenmaier J (2013) Framing image description as a ranking task: data, models and evaluation metrics. J Artif Intell Res 47:853–899

Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991

Hu Z, Zhang Z, Yang H, Chen Q, Zuo D (2017) A deep learning approach for predicting the quality of online health expert question-answering services. J Biomed Inform 71:241–253

Hu Z, Zhang Z, Yang H, Chen Q, Zhu R, Zuo D (2018) Predicting the quality of online health expert question-answering services with temporal features in a deep learning framework. Neurocomputing 275:2769–2782

Jean S, Cho K, Memisevic R, Bengio Y (2015) On using very large target vocabulary for neural machine translation. In: ACL, Beijing, pp 1–10
Ji S, Vishwanathan SVN, Satish N, Anderson JM, Dubey P (2015) Blackout: speeding up recurrent neural network language models with very large vocabularies. *CoRR*, abs/1511.06909
Johnson M, Schuster M, Le VQ, Krikun M, Wu Y, Chen Z, Thorat N, Viégas FB, Wattenberg M, Corrado G, Hughes M, Dean J (2017) Google's multilingual neural machine translation system: enabling zero-shot translation. TACL 5:339–351
Józefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In: ICML, Lille, pp 2342–2350
Junczys-Dowmunt M, Grundkiewicz R (2016) Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In: EMNLP, Austin, pp 1546–1556
Jurafsky D, Martin HJ (2017) Speech and language processing – an introduction to natural language processing. Computational linguistics, and speech recognition. 3rd edn. Prentice Hall, p 1032
Karpathy A, Fei-Fei L (2017) Deep visual-semantic alignments for generating image descriptions. IEEE Trans Pattern Anal Mach Intell 39(4):664–676
Kim Y (2014) Convolutional neural networks for sentence classification. In: EMNLP, Doha, pp 1746–1751
Kingma PD, Ba J (2014) Adam: a method for stochastic optimization. *CoRR*, abs/1412.6980
Koehn P (2005) Europarl: a parallel corpus for statistical machine translation. In: MT summit, vol 5, pp 79–86
Koutník J, Greff K, Gomez JF, Schmidhuber J (2014) A clockwork RNN. In: ICML, Beijing, pp 1863–1871
Krizhevsky A, Sutskever I, Hinton EG (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
Lafferty DJ, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML, Williams College, pp 282–289
Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: NAACL HLT, San Diego, pp 260–270
Lample G, Denoyer L, Ranzato M (2017) Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043
LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
LeCun Y, Bengio Y, Hinton EG (2015) Deep learning. Nature 521(7553):436–444
Lewis DD, Yang Y, Rose GT, Li F (2004) RCV1: a new benchmark collection for text categorization research. J Mach Learn Res 5:361–397
Lin T, Maire M, Belongie JS, Hays J, Perona P, Ramanan D, Dollár P, Zitnick LC (2014) Microsoft COCO: common objects in context. In: ECCV, Zurich, pp 740–755
Luong M, Manning DC (2016) Achieving open vocabulary neural machine translation with hybrid word-character models. In: ACL, Berlin
Luong M, Le VQ, Sutskever I, Vinyals O, Kaiser L (2015a) Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114
Luong T, Pham H, Manning DC (2015b) Effective approaches to attention-based neural machine translation. In: EMNLP, Lisbon, pp 1412–1421
Luong T, Sutskever I, Le VQ, Vinyals O, Zaremba W (2015c) Addressing the rare word problem in neural machine translation. In: ACL, Beijing, pp 11–19
Ma X, Hovy HE (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: ACL, Berlin

Maas LA, Daly ER, Pham TP, Huang D, Ng YA, Potts C (2011) Learning word vectors for sentiment analysis. In: The 49th annual meeting of the Association for Computational Linguistics: human language technologies, proceedings of the conference, 19–24 June 2011, Portland, pp 142–150

Manchanda B, Athavale AV, Kumar Sharma S (2016) Various techniques used for grammar checking. Int J Comput Appl Inf Technol 9(1):177

Manning DC (2011) Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: CICLing, Tokyo, pp 171–189

Marcus PM, Santorini B, Marcinkiewicz AM (1993) Building a large annotated corpus of English: the penn treebank. Comput Linguist 19(2):313–330

Marcus M, Santorini B, Marcinkiewicz M, Taylor A (1999) Treebank-3 LDC99T42. Web Download. Linguistic Data Consortium, Philadelphia. https://catalog.ldc.upenn.edu/LDC99T42

McCallum A, Freitag D, Pereira FCN (2000) Maximum entropy Markov models for information extraction and segmentation. In: ICML'00. Morgan Kaufmann Publishers Inc., San Francisco, pp 591–598

Melamud O, McClosky D, Patwardhan S, Bansal M (2016) The role of context types and dimensionality in learning word embeddings. In: NAACL HLT, San Diego, pp 1030–1040

Mi H, Wang Z, Ittycheriah A (2016) Vocabulary manipulation for neural machine translation. In: ACL, Berlin

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781

Mikolov T, Sutskever I, Chen K, Corrado SG, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS, Lake Tahoe, pp 3111–3119

Nazar R, Renau I (2012) Google books n-gram corpus used as a grammar checker. In: Proceedings of the second workshop on computational linguistics and writing (CLW 2012): linguistic and cognitive aspects of document creation and document engineering, EACL 2012, Stroudsburg. Association for Computational Linguistics, pp 27–34

Ng TH, Wu MS, Wu Y, Hadiwinoto C, Tetreault RJ (2013) The conll-2013 shared task on grammatical error correction. In: Proceedings of the seventeenth conference on computational natural language learning: shared task, CoNLL 2013, Sofia, 8–9 Aug 2013, pp 1–12

Ng TH, Wu MS, Briscoe T, Hadiwinoto C, Susanto HR, Bryant C (2014) The conll-2014 shared task on grammatical error correction. In: CoNLL, Baltimore, pp 1–14

Nivre J et al (2017) Universal dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (
'UFAL), Faculty of Mathematics and Physics, Charles University

Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd annual meeting of the Association for Computational Linguistics, Barcelona, 21–26 July 2004, pp 271–278

Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL 2005, 43rd annual meeting of the Association for Computational Linguistics, proceedings of the conference, 25–30 June 2005, University of Michigan, USA, pp 115–124

Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. *CoRR*, cs.CL/0205070

Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: ICML, Atlanta, pp 1310–1318

Pennington J, Socher R, Manning DC (2014) Glove: global vectors for word representation. In: EMNLP, Doha, pp 1532–1543

Plank B, Søgaard A, Goldberg Y (2016) Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In: ACL, Berlin

Plummer AB, Wang L, Cervantes MC, Caicedo CJ, Hockenmaier J, Lazebnik S (2017) Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. Int J Comput Vis 123(1):74–93

Rong X (2014) word2vec parameter learning explained. *CoRR*, abs/1411.2738

Rozovskaya A, Roth D (2010) Training paradigms for correcting errors in grammar and usage. In: HLT'10, Stroudsburg. Association for Computational Linguistics, pp 154–162

Rozovskaya A, Roth D (2016) Grammatical error correction: machine translation and classifiers. In: *ACL*, Berlin

Ruder S (2016) An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747

Ruder S, Ghaffari P, Breslin GJ (2016) A hierarchical model of reviews for aspect-based sentiment analysis. In: EMNLP, Austin, pp 999–1005

Schmaltz A, Kim Y, Rush MA, Shieber MS (2016) Sentence-level grammatical error identification as sequence-to-sequence correction. In: Proceedings of the 11th workshop on innovative use of NLP for building educational applications, BEA@NAACL-HLT 2016, 16 June 2016, San Diego, pp 242–251

Sennrich R, Haddow B, Birch A (2016a) Improving neural machine translation models with monolingual data. In: ACL, Berlin

Sennrich R, Haddow B, Birch A (2016b) Neural machine translation of rare words with subword units. In: ACL, Berlin

Srivastava N, Hinton EG, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Sutskever I, Vinyals O, Le VQ (2014) Sequence to sequence learning with neural networks. In: NIPS, Montreal, pp 3104–3112

Toutanova K, Klein D, Manning DC, Singer Y (2003) Feature-rich part-of-speech tagging with a cyclic dependency network. In: HLT-NAACL, Edmonton

Ueffing N, Ney H (2003) Using POS information for statistical machine translation into morphologically rich languages. In: EACL'03, Stroudsburg. Association for Computational Linguistics, pp 347–354

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez NA, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS, Long Beach, pp 6000–6010

Vinyals O, Toshev A, Bengio S, Erhan D (2017) Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. IEEE Trans Pattern Anal Mach Intell 39(4):652–663

Wang P, Qian Y, Soong KF, He L, Zhao H (2015) A unified tagging solution: bidirectional LSTM recurrent neural network with word embedding. *CoRR*, abs/1511.00215

Wiseman S, Rush MA (2016) Sequence-to-sequence learning as beam-search optimization. In: EMNLP, Austin, pp 1296–1306

Wu J, Chang J, Chang SJ (2013) Correcting serial grammatical errors based on n-grams and syntax. IJCLCLP 18(4)

Wu Y, Schuster M, Chen Z, Le VQ, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, Klingner J, Shah A, Johnson M, Liu X, Kaiser L, Gouws S, Kato Y, Kudo T, Kazawa H, Stevens K, Kurian G, Patil N, Wang W, Young C, Smith J, Riesa J, Rudnick A, Vinyals O, Corrado G, Hughes M, Dean J (2016) Google's neural machine translation system: bridging the gap between human and machine translation. *CoRR*, abs/1609.08144

Xu K, Ba J, Kiros R, Cho K, Courville CA, Salakhutdinov R, Zemel SR, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: ICML, Lille, pp 2048–2057

Yang Z, Salakhutdinov R, Cohen WW (2016) Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270

Yin W, Yu M, Xiang B, Zhou B, Schütze H (2016) Simple question answering by attentive convolutional neural network. In: COLING, Osaka, pp 1746–1756

Zhou J, Cao Y, Wang X, Li P, Xu W (2016) Deep recurrent models with fast-forward connections for neural machine translation. TACL 4:371–383

Ziemski M, Junczys-Dowmunt M, Pouliquen B (2016) The united nations parallel corpus v1.0. In: Proceedings of the tenth international conference on language resources and evaluation LREC 2016, Portorovz, 23–28 May 2016