Analysis Methods in Neural Language Processing: A Survey

Yonatan Belinkov\textsuperscript{1,2} and James Glass\textsuperscript{1}

\textsuperscript{1}MIT Computer Science and Artificial Intelligence Laboratory
\textsuperscript{2}Harvard School of Engineering and Applied Sciences
Cambridge, MA, USA
\{belinkov, glass\}@mit.edu

Abstract

The field of natural language processing has seen impressive progress in recent years, with neural network models replacing many of the traditional systems. A plethora of new models have been proposed, many of which are thought to be opaque compared to their feature-rich counterparts. This has led researchers to analyze, interpret, and evaluate neural networks in novel and more fine-grained ways. In this survey paper, we review analysis methods in neural language processing, categorize them according to prominent research trends, highlight existing limitations, and point to potential directions for future work.

1 Introduction

The rise of deep learning has transformed the field of natural language processing (NLP) in recent years. Models based on neural networks have obtained impressive improvements in various tasks, including language modeling (Mikolov et al., 2010; Jozefowicz et al., 2016), syntactic parsing (Kiperwasser and Goldberg, 2016), machine translation (MT) (Bahdanau et al., 2014; Sutskever et al., 2014), and many other tasks; see Goldberg (2017) for example success stories.

This progress has been accompanied by a myriad of new neural network architectures. In many cases, traditional feature-rich systems are being replaced by end-to-end neural networks that aim to map input text to some output prediction. As end-to-end systems are gaining prevalence, one may point to two trends. First, some push back against the abandonment of linguistic knowledge and call for incorporating it inside the networks in different ways.\textsuperscript{1} Others strive to better understand how NLP models work. This theme of analyzing neural networks has connections to the broader work on interpretability in machine learning, along with specific characteristics of the NLP field.

Why should we analyze our neural NLP models? To some extent, this question falls into the larger question of interpretability in machine learning, which has been the subject of much debate in recent years.\textsuperscript{2} Arguments in favor of interpretability in machine learning usually mention goals like accountability, trust, fairness, safety, and reliability (Doshi-Velez and Kim, 2017; Lipton, 2016). Arguments against interpretability typically stress performance as the most important desideratum. All these arguments naturally apply to machine learning applications in NLP.

In the context of NLP, this question needs to be understood in light of earlier NLP work, often referred to as feature-rich or feature-engineered systems. In some of these systems, features are more easily understood by humans—they can be morphological properties, lexical classes, syntactic categories, semantic relations, etc. In theory, one could observe the importance assigned by statistical NLP models to such features in order to gain a better understanding of the model.\textsuperscript{3} In

\textsuperscript{1}See, for instance, Noah Smith's invited talk at ACL 2017: vimeo.com/234958746. See also a recent debate on this matter by Chris Manning and Yann LeCun: www.youtube.com/watch?v=fKk9KhGRBdI. (Videos accessed on December 11, 2018.)

\textsuperscript{2}See, for example, the NIPS 2017 debate: www.youtube.com/watch?v=2hW05ZfsUuo. (Accessed on December 11, 2018.)

\textsuperscript{3}Nevertheless, one could question how feasible such an analysis is; consider, for example, interpreting support vectors in high-dimensional support vector machines (SVMs).
contrast, it is more difficult to understand what happens in an end-to-end neural network model that takes input (say, word embeddings) and generates an output (say, a sentence classification). Much of the analysis work thus aims to understand how linguistic concepts that were common as features in NLP systems are captured in neural networks.

As the analysis of neural networks for language is becoming more and more prevalent, neural networks in various NLP tasks are being analyzed; different network architectures and components are being compared, and a variety of new analysis methods are being developed. This survey aims to review and summarize this body of work, highlight current trends, and point to existing lacunae. It organizes the literature into several themes. Section 2 reviews work that targets a fundamental question: What kind of linguistic information is captured in neural networks? We also point to limitations in current methods for answering this question. Section 3 discusses visualization methods, and emphasizes the difficulty in evaluating visualization work. In Section 4, we discuss the compilation of challenge sets, or test suites, for fine-grained evaluation, a methodology that has old roots in NLP. Section 5 deals with the generation and use of adversarial examples to probe weaknesses of neural networks. We point to unique characteristics of dealing with text as a discrete input and how different studies handle them. Section 6 summarizes work on explaining model predictions, an important goal of interpretability research. This is a relatively underexplored area, and we call for more work in this direction. Section 7 mentions a few other methods that do not fall neatly into one of the above themes. In the conclusion, we summarize the main gaps and potential research directions for the field.

The paper is accompanied by online supplementary materials that contain detailed references for studies corresponding to Sections 2, 4, and 5 (Tables SM1, SM2, and SM3, respectively), available at https://boknilev.github.io/nlp-analysis-methods.

Before proceeding, we briefly mention some earlier work of a similar spirit.

**A Historical Note** Reviewing the vast literature on neural networks for language is beyond our scope. However, we mention here a few representative studies that focused on analyzing such networks in order to illustrate how recent trends have roots that go back to before the recent deep learning revival.

Rumelhart and McClelland (1986) built a feedforward neural network for learning the English past tense and analyzed its performance on a variety of examples and conditions. They were especially concerned with the performance over the course of training, as their goal was to model the past form acquisition in children. They also analyzed a scaled-down version having eight input units and eight output units, which allowed them to describe it exhaustively and examine how certain rules manifest in network weights.

In his seminal work on recurrent neural networks (RNNs), Elman trained networks on synthetic sentences in a language prediction task (Elman, 1989, 1990, 1991). Through extensive analyses, he showed how networks discover the notion of a word when predicting characters; capture syntactic structures like number agreement; and acquire word representations that reflect lexical and syntactic categories. Similar analyses were later applied to other networks and tasks (Harris, 1990; Niklasson and Linåker, 2000; Pollack, 1990; Frank et al., 2013).

While Elman’s work was limited in some ways, such as evaluating generalization or various linguistic phenomena—as Elman himself recognized (Elman, 1989)—it introduced methods that are still relevant today: from visualizing network activations in time, through clustering words by hidden state activations, to projecting representations to dimensions that emerge as capturing properties like sentence number or verb valency. The sections on visualization (Section 3) and identifying linguistic information (Section 2) contain many examples for these kinds of analysis.

## 2 What Linguistic Information Is Captured in Neural Networks?

Neural network models in NLP are typically trained in an end-to-end manner on input–output pairs, without explicitly encoding linguistic
features. Thus, a primary question is the following: What linguistic information is captured in neural networks? When examining answers to this question, it is convenient to consider three dimensions: which methods are used for conducting the analysis, what kind of linguistic information is sought, and which objects in the neural network are being investigated. Table SM1 (in the supplementary materials) categorizes relevant analysis work according to these criteria. In the next subsections, we discuss trends in analysis work along these lines, followed by a discussion of limitations of current approaches.

2.1 Methods

The most common approach for associating neural network components with linguistic properties is to predict such properties from activations of the neural network. Typically, in this approach a neural network model is trained on some task (say, MT) and its weights are frozen. Then, the trained model is used for generating feature representations for another task by running it on a corpus with linguistic annotations and recording the representations (say, hidden state activations). Another classifier is then used for predicting the property of interest (say, part-of-speech [POS] tags). The performance of this classifier is used for evaluating the quality of the generated representations, and by proxy that of the original model. This kind of approach has been used in numerous papers in recent years; see Table SM1 for references.\(^5\)

\(^5\)A similar method has been used to analyze hierarchical structure in neural networks trained on arithmetic expressions (Veldhoen et al., 2016; Hupkes et al., 2018).

It is referred to by various names, including “auxiliary prediction tasks” (Adi et al., 2017b), “diagnostic classifiers” (Veldhoen et al., 2016), and “probing tasks” (Conneau et al., 2018).

As an example of this approach, let us walk through an application to analyzing syntax in neural machine translation (NMT) by Shi et al. (2016b). In this work, two NMT models were trained on standard parallel data—English → French and English → German. The trained models (specifically, the encoders) were run on an annotated corpus and their hidden states were used for training a logistic regression classifier that predicts different syntactic properties. The authors concluded that the NMT encoders learn significant syntactic information at both word level and sentence level. They also compared representations at different encoding layers and found that “local features are somehow preserved in the lower layer whereas more global, abstract information tends to be stored in the upper layer.” These results demonstrate the kind of insights that the classification analysis may lead to, especially when comparing different models or model components.

Other methods for finding correspondences between parts of the neural network and certain properties include counting how often attention weights agree with a linguistic property like anaphora resolution (Voita et al., 2018) or directly computing correlations between neural network activations and some property; for example, correlating RNN state activations with depth in a syntactic tree (Qian et al., 2016a) or with Melfrequency cepstral coefficient (MFCC) acoustic features (Wu and King, 2016). Such correspondence may also be computed indirectly. For instance, Alishahi et al. (2017) defined an ABX discrimination task to evaluate how a neural model of speech (grounded in vision) encoded phonology. Given phoneme representations from different layers in their model, and three phonemes, A, B, and X, they compared whether the model representation for X is closer to A or B. This discrimination task enabled them to draw conclusions about which layers encoder phonology better, observing that lower layers generally encode more phonological information.

2.2 Linguistic Phenomena

Different kinds of linguistic information have been analyzed, ranging from basic properties like sentence length, word position, word presence, or simple word order, to morphological, syntactic, and semantic information. Phonetic/phonemic information, speaker information, and style and accent information have been studied in neural network models for speech, or in joint audio-visual models. See Table SM1 for references.

While it is difficult to synthesize a holistic picture from this diverse body of work, it appears that neural networks are able to learn a substantial amount of information on various linguistic phenomena. These models are especially successful at capturing frequent properties, while some rare properties are more difficult to learn.
Linzen et al. (2016), for instance, found that long short-term memory (LSTM) language models are able to capture subject–verb agreement in many common cases, while direct supervision is required for solving harder cases.

Another theme that emerges in several studies is the hierarchical nature of the learned representations. We have already mentioned such findings regarding NMT (Shi et al., 2016b) and a visually grounded speech model (Alishahi et al., 2017). Hierarchical representations of syntax were also reported to emerge in other RNN models (Blevins et al., 2018).

Finally, a couple of papers discovered that models trained with latent trees perform better on natural language inference (NLI) (Williams et al., 2018; Maillard and Clark, 2018) than ones trained with linguistically annotated trees. Moreover, the trees in these models do not resemble syntactic trees corresponding to known linguistic theories, which casts doubts on the importance of syntax-learning in the underlying neural network.  

2.3 Neural Network Components

In terms of the object of study, various neural network components were investigated, including word embeddings, RNN hidden states or gate activations, sentence embeddings, and attention weights in sequence-to-sequence (seq2seq) models. Generally less work has analyzed convolutional neural networks in NLP, but see Jacovi et al. (2018) for a recent exception. In speech processing, researchers have analyzed layers in deep neural networks for speech recognition and different speaker embeddings. Some analysis has also been devoted to joint language–vision or audio–vision models, or to similarities between word embeddings and convolutional image representations. Table SM1 provides detailed references.

2.4 Limitations

The classification approach may find that a certain amount of linguistic information is captured in the neural network. However, this does not necessarily mean that the information is used by the network. For example, Vanmassenhove et al. (2017) investigated aspect in NMT (and in phrase-based statistical MT). They trained a classifier on NMT sentence encoding vectors and found that they can accurately predict tense about 90% of the time. However, when evaluating the output translations, they found them to have the correct tense only 79% of the time. They interpreted this result to mean that “part of the aspectual information is lost during decoding.” Relatedly, Cifka and Bojar (2018) compared the performance of various NMT models in terms of translation quality (BLEU) and representation quality (classification tasks). They found a negative correlation between the two, suggesting that high-quality systems may not be learning certain sentence meanings. In contrast, Artetxe et al. (2018) showed that word embeddings contain divergent linguistic information, which can be uncovered by applying a linear transformation on the learned embeddings. Their results suggest an alternative explanation, showing that “embedding models are able to encode divergent linguistic information but have limits on how this information is surfaced.”

From a methodological point of view, most of the relevant analysis work is concerned with correlation: How correlated are neural network components with linguistic properties? What may be lacking is a measure of causation: How does the encoding of linguistic properties affect the system output? Giulianelli et al. (2018) make some headway on this question. They predicted number agreement from RNN hidden states and gates at different time steps. They then intervened in how the model processes the sentence by changing a hidden activation based on the difference between the prediction and the correct label. This improved agreement prediction accuracy, and the effect persisted over the course of the sentence, indicating that this information has an effect on the model. However, they did not report the effect on overall model quality, for example by measuring perplexity. Methods from causal inference may shed new light on some of these questions.

Finally, the predictor for the auxiliary task is usually a simple classifier, such as logistic regression. A few studies compared different classifiers and found that deeper classifiers lead to overall better results, but do not alter the respective trends when comparing different models or components (Qian et al., 2016b; Belinkov, 2018). Interestingly, Conneau et al. (2018) found that tasks requiring more nuanced linguistic knowledge

---

6 Others found that even simple binary trees may work well in MT (Wang et al., 2018b) and sentence classification (Chen et al., 2015).
(e.g., tree depth, coordination inversion) gain the most from using a deeper classifier. However, the approach is usually taken for granted; given its prevalence, it appears that better theoretical or empirical foundations are in place.

3 Visualization

Visualization is a valuable tool for analyzing neural networks in the language domain and beyond. Early work visualized hidden unit activations in RNNs trained on an artificial language modeling task, and observed how they correspond to certain grammatical relations such as agreement (Elman, 1991). Much recent work has focused on visualizing activations on specific examples in modern neural networks for language (Karpathy et al., 2015; Kádár et al., 2017; Qian et al., 2016a; Liu et al., 2018) and speech (Wu and King, 2016; Nagamine et al., 2015; Wang et al., 2017b). Figure 1 shows an example visualization of a neuron that captures position of words in a sentence. The heatmap uses blue and red colors for negative and positive activation values, respectively, enabling the user to quickly grasp the function of this neuron.

The attention mechanism that originated in work on NMT (Bahdanau et al., 2014) also lends itself to a natural visualization. The alignments obtained via different attention mechanisms have produced visualizations ranging from tasks like NLI (Rocktäschel et al., 2016; Yin et al., 2016), summarization (Rush et al., 2015), MT post-editing (Jauregi Unanue et al., 2018), and morphological inflection (Aharoni and Goldberg, 2017) to matching users on social media (Tay et al., 2018). Figure 2 reproduces a visualization of attention alignments from the original work by Bahdanau et al. Here grayscale values correspond to the weight of the attention between words in an English source sentence (columns) and its French translation (rows). As Bahdanau et al. explain, this visualization demonstrates that the NMT model learned a soft alignment between source and target words. Some aspects of word order may also be noticed, as in the reordering of noun and adjective when translating the phrase “European Economic Area.”

Another line of work computes various saliency measures to attribute predictions to input features. The important or salient features can then be visualized in selected examples (Li et al., 2016a; Aubakirova and Bansal, 2016; Sundararajan et al., 2017; Arras et al., 2017a,b; Ding et al., 2017; Murdoch et al., 2018; Mudrakarta et al., 2018; Montavon et al., 2018; Godin et al., 2018). Saliency can also be computed with respect to intermediate values, rather than input features (Ghaeini et al., 2018).

An instructive visualization technique is to cluster neural network activations and compare them to some linguistic property. Early work clustered RNN activations, showing that they organize in lexical categories (Elman, 1989, 1990). Similar techniques have been followed by others. Recent examples include clustering of sentence embeddings in an RNN encoder trained in a multitask learning scenario (Brunner et al., 2017), and phoneme clusters in a joint audio-visual RNN model (Alishahi et al., 2017).

A few online tools for visualizing neural networks have recently become available. LSTMVis generally, many of the visualization methods are adapted from the vision domain, where they have been extremely popular; see Zhang and Zhu (2018) for a survey.
(Strobelt et al., 2018b) visualizes RNN activations, focusing on tracing hidden state dynamics. Seq2Seq-Vis (Strobelt et al., 2018a) visualizes different modules in attention-based seq2seq models, with the goal of examining model decisions and testing alternative decisions. Another tool focused on comparing attention alignments was proposed by Rikters (2018). It also provides translation confidence scores based on the distribution of attention weights. NeuroX (Dalvi et al., 2019b) is a tool for finding and analyzing individual neurons, focusing on machine translation.

Evaluation As in much work on interpretability, evaluating visualization quality is difficult and often limited to qualitative examples. A few notable exceptions report human evaluations of visualization quality. Singh et al. (2018) showed human raters hierarchical clusterings of input words generated by two interpretation methods, and asked them to evaluate which method is more accurate, or in which method they trust more. Others reported human evaluations for attention visualization in conversation modeling (Freeman et al., 2018) and medical code prediction tasks (Mullenbach et al., 2018).

The availability of open-source tools of the sort described above will hopefully encourage users to utilize visualization in their regular research and development cycle. However, it remains to be seen how useful visualizations turn out to be.

4 Challenge Sets

The majority of benchmark datasets in NLP are drawn from text corpora, reflecting a natural frequency distribution of language phenomena. While useful in practice for evaluating system performance in the average case, such datasets may fail to capture a wide range of phenomena. An alternative evaluation framework consists of challenge sets, also known as test suites, which have been used in NLP for a long time (Lehmann et al., 1996), especially for evaluating MT systems (King and Falkedal, 1990; Isahara, 1995; Koh et al., 2001). Lehmann et al. (1996) noted several key properties of test suites: systematicity, control over data, inclusion of negative data, and exhaustivity. They contrasted such datasets with test corpora, “whose main advantage is that they reflect naturally occurring data.” This idea underlines much of the work on challenge sets and is echoed in more recent work (Wang et al., 2018a). For instance, Cooper et al. (1996) constructed a semantic test suite that targets phenomena as diverse as quantifiers, plurals, anaphora, ellipsis, adjectival properties, and so on.

After a hiatus of a couple of decades, challenge sets have recently gained renewed popularity in the NLP community. In this section, we include datasets used for evaluating neural network models that diverge from the common average-case evaluation. Many of them share some of the properties noted by Lehmann et al. (1996), although negative examples (ill-formed data) are typically less utilized. The challenge datasets can be categorized along the following criteria: the task they seek to evaluate, the linguistic phenomena they aim to study, the language(s) they target, their size, their method of construction, and how performance is evaluated. Table SM2 (in the supplementary materials) categorizes many recent challenge sets along these criteria. Below we discuss common trends along these lines.

4.1 Task

By far, the most targeted tasks in challenge sets are NLI and MT. This can partly be explained by the popularity of these tasks and the prevalence of neural models proposed for solving them. Perhaps more importantly, tasks like NLI and MT arguably require inferences at various linguistic levels, making the challenge set evaluation especially attractive. Still, other high-level tasks like reading comprehension or question answering have not received as much attention, and may also benefit from the careful construction of challenge sets.

A significant body of work aims to evaluate the quality of embedding models by correlating the similarity they induce on word or sentence pairs with human similarity judgments. Datasets containing such similarity scores are often used

---

8RNNVis (Ming et al., 2017) is a similar tool, but its online demo does not seem to be available at the time of writing.

9One could speculate that their decrease in popularity can be attributed to the rise of large-scale quantitative evaluation of statistical NLP systems.

10Another typology of evaluation protocols was put forth by Burlot and Yvon (2017). Their criteria are partially overlapping with ours, although they did not provide a comprehensive categorization like the one compiled here.
to evaluate word embeddings (Finkelstein et al., 2002; Bruni et al., 2012; Hill et al., 2015, inter alia) or sentence embeddings; see the many shared tasks on semantic textual similarity in SemEval (Cer et al., 2017, and previous editions). Many of these datasets evaluate similarity at a coarse-grained level, but some provide a more fine-grained evaluation of similarity or relatedness. For example, some datasets are dedicated for specific word classes such as verbs (Gerz et al., 2016) or rare words (Luong et al., 2013), or for evaluating compositional knowledge in sentence embeddings (Marelli et al., 2014). Multilingual and cross-lingual versions have also been collected (Leviant and Reichart, 2015; Cer et al., 2017). Although these datasets are widely used, this kind of evaluation has been criticized for its subjectivity and questionable correlation with downstream performance (Faruqui et al., 2016).

4.2 Linguistic Phenomena

One of the primary goals of challenge sets is to evaluate models on their ability to handle specific linguistic phenomena. While earlier studies emphasized exhaustivity (Cooper et al., 1996; Lehmann et al., 1996), recent ones tend to focus on a few properties of interest. For example, Sennrich (2017) introduced a challenge set for MT evaluation focusing on five properties: subject–verb agreement, noun phrase agreement, verb–particle constructions, polarity, and transliteration. Slightly more elaborated is an MT challenge set for morphology, including 14 morphological properties (Burlot and Yvon, 2017). See Table SM2 for references to datasets targeting other phenomena.

Other challenge sets cover a more diverse range of linguistic properties, in the spirit of some of the earlier work. For instance, extending the categories in Cooper et al. (1996), the GLUE analysis set for NLI covers more than 30 phenomena in four coarse categories (lexical semantics, predicate–argument structure, logic, and knowledge). In MT evaluation, Burchardt et al. (2017) reported results using a large test suite covering 120 phenomena, partly based on Lehmann et al. (1996).11 Isabelle et al. (2017) and Isabelle and Kuhn (2018) prepared challenge sets for MT evaluation covering fine-grained phenomena at morpho-syntactic, syntactic, and lexical levels.

Generally, datasets that are constructed programmatically tend to cover less fine-grained linguistic properties, while manually constructed datasets represent more diverse phenomena.

4.3 Languages

As unfortunately usual in much NLP work, especially neural NLP, the vast majority of challenge sets are in English. This situation is slightly better in MT evaluation, where naturally all datasets feature other languages (see Table SM2). A notable exception is the work by Gulordava et al. (2018), who constructed examples for evaluating number agreement in language modeling in English, Russian, Hebrew, and Italian. Clearly, there is room for more challenge sets in non-English languages. However, perhaps more pressing is the need for large-scale non-English datasets (besides MT) to develop neural models for popular NLP tasks.

4.4 Scale

The size of proposed challenge sets varies greatly (Table SM2). As expected, datasets constructed by hand are smaller, with typical sizes in the hundreds. Automatically built datasets are much larger, ranging from several thousands to close to a hundred thousand (Sennrich, 2017), or even more than one million examples (Linzen et al., 2016). In the latter case, the authors argue that such a large test set is needed for obtaining a sufficient representation of rare cases. A few manually constructed datasets contain a fairly large number of examples, up to 10 thousand (Burchardt et al., 2017).

4.5 Construction Method

Challenge sets are usually created either programmatically or manually, by handcrafting specific examples. Often, semi-automatic methods are used to compile an initial list of examples that is manually verified by annotators. The specific method also affects the kind of language use and how natural or artificial/synthetic the examples are. We describe here some trends in dataset construction methods in the hope that they may be useful for researchers contemplating new datasets.
Several datasets were constructed by modifying or extracting examples from existing datasets. For instance, Sanchez et al. (2018) and Glockner et al. (2018) extracted examples from SNLI (Bowman et al., 2015) and replaced specific words such as hypernyms, synonyms, and antonyms, followed by manual verification. Linzen et al. (2016), on the other hand, extracted examples of subject–verb agreement from raw texts using heuristics, resulting in a large-scale dataset. Gulordava et al. (2018) extended this to other agreement phenomena, but they relied on syntactic information available in treebanks, resulting in a smaller dataset.

Several challenge sets utilize existing test suites, either as a direct source of examples (Burchardt et al., 2017) or for searching similar naturally occurring examples (Wang et al., 2018a).

Sennrich (2017) introduced a method for evaluating NMT systems via contrastive translation pairs, where the system is asked to estimate the probability of two candidate translations that are designed to reflect specific linguistic properties. Sennrich generated such pairs programatically by applying simple heuristics, such as changing gender and number to induce agreement errors, resulting in a large-scale challenge set of close to 100 thousand examples. This framework was extended to evaluate other properties, but often requiring more sophisticated generation methods like using morphological analyzers/generators (Burlot and Yvon, 2017) or more manual involvement in generation (Bawden et al., 2018) or verification (Rios Gonzales et al., 2017).

Finally, a few studies define templates that capture certain linguistic properties and instantiate them with word lists (Dasgupta et al., 2018; Rudinger et al., 2018; Zhao et al., 2018a). Template-based generation has the advantage of providing more control, for example for obtaining a specific vocabulary distribution, but this comes at the expense of how natural the examples are.

4.6 Evaluation

Systems are typically evaluated by their performance on the challenge set examples, either with the same metric used for evaluating the system in the first place, or via a proxy, as in the contrastive pairs evaluation of Sennrich (2017). Automatic evaluation metrics are cheap to obtain and can be calculated on a large scale. However, they may miss certain aspects. Thus a few studies report human evaluation on their challenge sets, such as in MT (Isabelle et al., 2017; Burchardt et al., 2017).

We note here also that judging the quality of a model by its performance on a challenge set can be tricky. Some authors emphasize their wish to test systems on extreme or difficult cases, “beyond normal operational capacity” (Naik et al., 2018). However, whether one should expect systems to perform well on specially chosen cases (as opposed to the average case) may depend on one’s goals. To put results in perspective, one may compare model performance to human performance on the same task (Gulordava et al., 2018).

5 Adversarial Examples

Understanding a model also requires an understanding of its failures. Despite their success in many tasks, machine learning systems can also be very sensitive to malicious attacks or adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). In the vision domain, small changes to the input image can lead to misclassification, even if such changes are indistinguishable by humans.

The basic setup in work on adversarial examples can be described as follows. Given a neural network model $f$ and an input example $x$, we seek to generate an adversarial example $x'$ that will have a minimal distance from $x$, while being assigned a different label by $f$:

$$\min_{x'} ||x - x'|| \text{ s.t. } f(x) = l, f(x') = l', l \neq l'$$

In the vision domain, $x$ can be the input image pixels, resulting in a fairly intuitive interpretation of this optimization problem: measuring the distance $||x - x'||$ is straightforward, and finding $x'$ can be done by computing gradients with respect to the input, since all quantities are continuous.

In the text domain, the input is discrete (for example, a sequence of words), which poses two problems. First, it is not clear how to measure

\[\text{12} \text{Wang et al. (2018a) also verified that their examples do not contain annotation artifacts, a potential problem noted in recent studies (Gururangan et al., 2018; Poliak et al., 2018b).} \]

\[\text{13} \text{The notation here follows Yuan et al. (2017).} \]
the distance between the original and adversarial examples, \( x \) and \( x' \), which are two discrete objects (say, two words or sentences). Second, minimizing this distance cannot be easily formulated as an optimization problem, as this requires computing gradients with respect to a discrete input.

In the following, we review methods for handling these difficulties according to several criteria: the adversary’s knowledge, the specificity of the attack, the linguistic unit being modified, and the task on which the attacked model was trained.\(^\text{14}\) Table SM3 (in the supplementary materials) categorizes work on adversarial examples in NLP according to these criteria.

### 5.1 Adversary’s Knowledge

Adversarial examples can be generated using access to model parameters, also known as white-box attacks, or without such access, with black-box attacks (Papernot et al., 2016a, 2017; Narodytska and Kasiviswanathan, 2017; Liu et al., 2017).

White-box attacks are difficult to adapt to the text world as they typically require computing gradients with respect to the input, which would be discrete in the text case. One option is to compute gradients with respect to the input word embeddings, and perturb the embeddings. Since this may result in a vector that does not correspond to any word, one could search for the closest word embedding in a given dictionary (Papernot et al., 2016b); Cheng et al. (2018) extended this idea to seq2seq models. Others computed gradients with respect to input word embeddings to identify and rank words to be modified (Samanta and Mehta, 2017; Liang et al., 2018). Ebrahimi et al. (2018b) developed an alternative method by representing text edit operations in vector space (e.g., a binary vector specifying which characters in a word would be changed) and approximating the change in loss with the derivative along this vector.

Given the difficulty in generating white-box adversarial examples for text, much research has been devoted to black-box examples. Often, the adversarial examples are inspired by text edits that are thought to be natural or commonly generated by humans, such as typos, misspellings, and so on (Sakaguchi et al., 2017; Heigold et al., 2018; Belinkov and Bisk, 2018). Gao et al. (2018) defined scoring functions to identify tokens to modify. Their functions do not require access to model internals, but they do require the model prediction score. After identifying the important tokens, they modify characters with common edit operations.

Zhao et al. (2018c) used generative adversarial networks (GANs) (Goodfellow et al., 2014) to minimize the distance between latent representations of input and adversarial examples, and performed perturbations in latent space. Since the latent representations do not need to come from the attacked model, this is a black-box attack.

Finally, Alzantot et al. (2018) developed an interesting population-based genetic algorithm for crafting adversarial examples for text classification by maintaining a population of modifications of the original sentence and evaluating fitness of modifications at each generation. They do not require access to model parameters, but do use prediction scores. A similar idea was proposed by Kuleshov et al. (2018).

### 5.2 Attack Specificity

Adversarial attacks can be classified to targeted vs. non-targeted attacks (Yuan et al., 2017). A targeted attack specifies a specific false class, \( l' \), while a nontargeted attack cares only that the predicted class is wrong, \( l' \neq l \). Targeted attacks are more difficult to generate, as they typically require knowledge of model parameters; that is, they are white-box attacks. This might explain why the majority of adversarial examples in NLP are nontargeted (see Table SM3). A few targeted attacks include Liang et al. (2018), which specified a desired class to fool a text classifier, and Chen et al. (2018a), which specified words or captions to generate in an image captioning model. Others targeted specific words to omit, replace, or include when attacking seq2seq models (Cheng et al., 2018; Ebrahimi et al., 2018a).

Methods for generating targeted attacks in NLP could possibly take more inspiration from adversarial attacks in other fields. For instance, in attacking malware detection systems, several studies developed targeted attacks in a black-box scenario (Yuan et al., 2017). A black-box targeted attack for MT was proposed by Zhao et al. (2018c), who used GANs to search for...
attacks on Google’s MT system after mapping sentences into continuous space with adversarially regularized autoencoders (Zhao et al., 2018b).

5.3 Linguistic Unit
Most of the work on adversarial text examples involves modifications at the character- and/or word-level; see Table SM3 for specific references. Other transformations include adding sentences or text chunks (Jia and Liang, 2017) or generating paraphrases with desired syntactic structures (Iyyer et al., 2018). In image captioning, Chen et al. (2018a) modified pixels in the input image to generate targeted attacks on the caption text.

5.4 Task
Generally, most work on adversarial examples in NLP concentrates on relatively high-level language understanding tasks, such as text classification (including sentiment analysis) and reading comprehension, while work on text generation focuses mainly on MT. See Table SM3 for references. There is relatively little work on adversarial examples for more low-level language processing tasks, although one can mention morphological tagging (Heigold et al., 2018) and spelling correction (Sakaguchi et al., 2017).

5.5 Coherence and Perturbation Measurement
In adversarial image examples, it is fairly straightforward to measure the perturbation, either by measuring distance in pixel space, say $\|x - x'\|$ under some norm, or with alternative measures that are better correlated with human perception (Rozsa et al., 2016). It is also visually compelling to present an adversarial image with imperceptible difference from its source image. In the text domain, measuring distance is not as straightforward, and even small changes to the text may be perceptible by humans. Thus, evaluation of attacks is fairly tricky. Some studies imposed constraints on adversarial examples to have a small number of edit operations (Gao et al., 2018). Others ensured syntactic or semantic coherence in different ways, such as filtering replacements by word similarity or sentence similarity (Alzantot et al., 2018; Kuleshov et al., 2018), or by using synonyms and other word lists (Samanta and Mehta, 2017; Yang et al., 2018).

Some reported whether a human can classify the adversarial example correctly (Yang et al., 2018), but this does not indicate how perceptible the changes are. More informative human studies evaluate grammaticality or similarity of the adversarial examples to the original ones (Zhao et al., 2018c; Alzantot et al., 2018). Given the inherent difficulty in generating imperceptible changes in text, more such evaluations are needed.

6 Explaining Predictions
Explaining specific predictions is recognized as a desideratum in interpretability work (Lipton, 2016), argued to increase the accountability of machine learning systems (Doshi-Velez et al., 2017). However, explaining why a deep, highly non-linear neural network makes a certain prediction is not trivial. One solution is to ask the model to generate explanations along with its primary prediction (Zaidan et al., 2007; Zhang et al., 2016), but this approach requires manual annotations of explanations, which may be hard to collect.

An alternative approach is to use parts of the input as explanations. For example, Lei et al. (2016) defined a generator that learns a distribution over text fragments as candidate rationales for justifying predictions, evaluated on sentiment analysis. Alvarez-Melis and Jaakkola (2017) discovered input–output associations in a sequence-to-sequence learning scenario, by perturbing the input and finding the most relevant associations. Gupta and Schütze (2018) inspected how information is accumulated in RNNs towards a prediction, and associated peaks in prediction scores with important input segments. As these methods use input segments to explain predictions, they do not shed much light on the internal computations that take place in the network.

At present, despite the recognized importance for interpretability, our ability to explain predictions of neural networks in NLP is still limited.

7 Other Methods
We briefly mention here several analysis methods that do not fall neatly into the previous sections.

A number of studies evaluated the effect of erasing or masking certain neural network components, such as word embedding dimensions, hidden units, or even full words (Li et al., 2016b; Li et al., 2016a; Other work considered learning textual-visual explanations from multimodal annotations (Park et al., 2018).
Feng et al., 2018; Khandelwal et al., 2018; Bau et al., 2018). For example, Li et al. (2016b) erased specific dimensions in word embeddings or hidden states and computed the change in probability assigned to different labels. Their experiments revealed interesting differences between word embedding models, where in some models information is more focused in individual dimensions. They also found that information is more distributed in hidden layers than in the input layer, and erased entire words to find important words in a sentiment analysis task.

Several studies conducted behavioral experiments to interpret word embeddings by defining intrusion tasks, where humans need to identify an intruder word, chosen based on difference in word embedding dimensions (Murphy et al., 2012; Fyshe et al., 2015; Faruqui et al., 2015). In this kind of work, a word embedding model may be deemed more interpretable if humans are better able to identify the intruding words. Since the evaluation is costly for high-dimensional representations, alternative automatic metrics were considered (Park et al., 2017; Senel et al., 2018).

A long tradition in work on neural networks is to evaluate and analyze their ability to learn different formal languages (Das et al., 1992; Casey, 1996; Gers and Schmidhuber, 2001; Bodén and Wiles, 2002; Chalup and Blair, 2003). This trend continues today, with research into modern architectures and what formal languages they can learn (Weiss et al., 2018; Bernardy, 2018; Suzgun et al., 2019), or the formal properties they possess (Chen et al., 2018b).

8 Conclusion

Analyzing neural networks has become a hot topic in NLP research. This survey attempted to review and summarize as much of the current research as possible, while organizing it along several prominent themes. We have emphasized aspects in analysis that are specific to language—namely, what linguistic information is captured in neural networks, which phenomena they are successful at capturing, and where they fail. Many of the analysis methods are general techniques from the larger machine learning community, such as visualization via saliency measures or evaluation by adversarial examples. But even those sometimes require non-trivial adaptations to work with text input. Some methods are more specific to the field, but may prove useful in other domains. Challenge sets or test suites are such a case.

Throughout this survey, we have identified several limitations or gaps in current analysis work:

- The use of auxiliary classification tasks for identifying which linguistic properties neural networks capture has become standard practice (Section 2), while lacking both a theoretical foundation and a better empirical consideration of the link between the auxiliary tasks and the original task.
- Evaluation of analysis work is often limited or qualitative, especially in visualization techniques (Section 3). Newer forms of evaluation are needed for determining the success of different methods.
- Relatively little work has been done on explaining predictions of neural network models, apart from providing visualizations (Section 6). With the increasing public demand for explaining algorithmic choices in machine learning systems (Doshi-Velez and Kim, 2017; Doshi-Velez et al., 2017), there is pressing need for progress in this direction.
- Much of the analysis work is focused on the English language, especially in constructing challenge sets for various tasks (Section 4), with the exception of MT due to its inherent multilingual character. Developing resources and evaluating methods on other languages is important as the field grows and matures.
- More challenge sets for evaluating other tasks besides NLI and MT are needed.

Finally, as with any survey in a rapidly evolving field, this paper is likely to omit relevant recent work by the time of publication. While we intend to continue updating the online appendix with newer publications, we hope that our summarization of prominent analysis work and its categorization into several themes will be a useful guide for scholars interested in analyzing and understanding neural networks for NLP.
Acknowledgments

We would like to thank the anonymous reviewers and the action editor for their very helpful comments. This work was supported by the Qatar Computing Research Institute. Y.B. is also supported by the Harvard Mind, Brain, Behavior Initiative.

References


Jean-Philippe Bernardy. 2018. Can Recurrent Neural Networks Learn Nested Recursion? LiLT (Linguistic Issues in Language Technology), 16(1).


Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of


Hitoshi Isahara. 1995. JEIDA’s test-sets for quality evaluation of MT systems—technical evaluation from the developer’s point of view. In Proceedings of MT Summit V.


Margaret King and Kirsten Falkedal. 1990. Using Test Suites in Evaluation of Machine


Conference on Computational Natural Language Learning, pages 104–113. Association for Computational Linguistics.


Xing Shi, Inkit Padhi, and Kevin Knight. 2016b. Does String-Based Neural MT Learn Source


