



On the Explainability of Natural Language Processing Deep Models

JULIA EL ZINI and MARIETTE AWAD, Department of Electrical and Computer Engineering,
American University of Beirut

Despite their success, deep networks are used as black-box models with outputs that are not easily explainable during the learning and the prediction phases. This lack of interpretability is significantly limiting the adoption of such models in domains where decisions are critical such as the medical and legal fields. Recently, researchers have been interested in developing methods that help explain individual decisions and decipher the hidden representations of machine learning models in general and deep networks specifically. While there has been a recent explosion of work on Explainable Artificial Intelligence (ExAI) on deep models that operate on imagery and tabular data, textual datasets present new challenges to the ExAI community. Such challenges can be attributed to the lack of input structure in textual data, the use of word embeddings that add to the opacity of the models and the difficulty of the visualization of the inner workings of deep models when they are trained on textual data.

Lately, methods have been developed to address the aforementioned challenges and present satisfactory explanations on Natural Language Processing (NLP) models. However, such methods are yet to be studied in a comprehensive framework where common challenges are properly stated and rigorous evaluation practices and metrics are proposed.

Motivated to democratize ExAI methods in the NLP field, we present in this work a survey that studies *model-agnostic* as well as *model-specific* explainability methods on NLP models. Such methods can either develop *inherently* interpretable NLP models or operate on pre-trained models in a *post hoc* manner. We make this distinction and we further decompose the methods into three categories according to what they explain: (1) word embeddings (input level), (2) inner workings of NLP models (processing level), and (3) models' decisions (output level). We also detail the different evaluation approaches interpretability methods in the NLP field. Finally, we present a case-study on the well-known neural machine translation in an appendix, and we propose promising future research directions for ExAI in the NLP field.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Machine learning**; **Artificial intelligence**; • **Machine learning** → **Neural machine translation**;

Additional Key Words and Phrases: ExAI, NLP, language models, transformers, neural machine translation, transparent embedding models, explaining decisions

ACM Reference format:

Julia El Zini and Mariette Awad. 2022. On the Explainability of Natural Language Processing Deep Models. *ACM Comput. Surv.* 55, 5, Article 103 (December 2022), 31 pages.
<https://doi.org/10.1145/3529755>

Authors' address: J. El Zini and M. Awad, Department of Electrical and Computer Engineering, American University of Beirut, P. O. Box 11-0236, Riad El Solh, Beirut, Lebanon, 1107-2020; emails: {jwe04, mariette.awad}@aub.edu.lb.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

0360-0300/2022/12-ART103 \$15.00

<https://doi.org/10.1145/3529755>

1 INTRODUCTION

Ever since their introduction, **deep learning (DL)** models are revolutionizing several **Natural Language Processing (NLP)** applications ranging from machine translation [8, 113, 119] to text summarization [107] and question answering [25, 121, 122]. AI-powered systems that mainly use DL models are reaching highly accurate predictions that surpass human performances in some cases [100]. However, due to their non-linear multilayered structure, deep networks are often seen as black-box models that achieve high performances but in an opaque manner. Researchers and practitioners often question how credible these predictions are if the reasoning behind them is a highly non-linear enigma that cannot be easily deciphered. This black-box nature of DL models gives rise to several criticisms of their non-transparent predictions. Transparency and interpretability are thus needed to establish user trust when black-box DL achieves a performance comparable to that of the human. Besides, weak DL models entail interpretability to investigate failure cases and direct the researchers in the proper paths [27]. The need for interpretability is even more pronounced when DL models beat human performance where explanations can serve as a machine teaching framework for the human to make better decisions. For instance, if the decision-making system of Alpha Go [100] that beat the world champion in the Go game was transparent, then some creative moves can be taught to humans to help them learn the game or even extend their mental capabilities.

Transparency is not only needed on the prediction level; some situations require DL models that achieve explainability during the learning phases. For instance, DL models that learn from curated datasets might engender bias that is not easy to detect requiring a higher explainability level [71]. Interpretability of the hidden representations of these networks and the understanding of the predictions of the whole category would suggest whether some protected attributes are affecting the predictions in a biased manner.

Recently, the **Explainable Artificial Intelligence (ExAI)** field has attracted researchers to develop explainability methods for black-box deep networks [63, 88, 98] in general. While some of these methods can be directly applied to NLP models [6, 63, 88], others are specific to imagery datasets or **Convolutional Neural Networks (CNNs)** that are not very suitable for neural machine understanding tasks [21, 30, 66, 98]. It is worth mentioning that explainability and interpretability are used interchangeably in this work. For definitions and terminologies, readers are referred to Reference [68].

We focus our survey on the interpretability of language models or deep networks that operate on textual data. We make the distinction between the methods that develop *inherently* interpretable models and those that operate in a *post hoc* manner. We also make the distinction between *model-specific* and *model-agnostic* methods and the level at which each method operates: input (or embedding), processing, and prediction level. Throughout this work, we use the terms *explainability* and *interpretability* interchangeably. For more details on definitions and terminologies, we refer readers to Reference [68]. We make the distinction between the methods that develop *inherently* interpretable models and those that operate in a *post hoc* manner. We also make the distinction between *model-specific* and *model-agnostic* methods and the level at which each method operates: input (or embedding), processing, and prediction level.

Although interpretability is a relatively new research track in AI, different surveys have been compiled to highlight specific interpretability aspects [28, 37, 70, 74, 85]. Recently, Reference [22] addressed interpretability of NLP models in their survey while focusing on explaining model's decision. Unlike Reference [22], this work is not only limited to ExAI methods on the decision level. Our work here contributes to the community the first survey on the (1) interpretability of word embeddings models that constitute a crucial part of NLP, (2) the inner representations of NLP

networks, and (3) the transformer models that presented a great debate on the interpretability of their attention mechanisms. In addition to that, we survey existing work on the explainability of individual model decisions with insights about research challenges and opportunities in that field. We also highlight different empirical setups, metrics, and datasets that NLP researchers rely on to evaluate their ExAI methods.

The rest of this article is organized as follows: Section 2 presents the challenges that general ExAI methods face with textual datasets or NLP models, and Section 3 presents the related surveys to ExAI. Then, Section 4 presents the terminology used in this article with respect to the three proposed dimensions. The rest of the sections are focused on the interpretability of NLP models on three levels: the input, the processing, and the output. Section 5 surveys existing work on the interpretability of word embeddings serving as inputs to NLP models; Sections 6 and 7 present the interpretability methods applied to the inner representations of **Recurrent Neural Networks (RNNs)** and transformers, with their debatable attention mechanisms, respectively, during the processing phase; and Section 8 focuses on the explanations of individual predictions or outputs. Finally, we end with some concluding remarks and future directions in Sections 9 and 10 while presenting a case study on neural machine translation in an appendix.

2 CHALLENGES IN EXAI ON NLP MODELS

In the traditional learning framework, textual datasets present many challenges such as polysemy, sarcasm, slang, cultural effect, and ambiguity. Explainability methods proliferate these challenges. In what follows, we identify some text-specific challenges that hamper the application of general ExAI methods into NLP models. Those challenges motivate the necessity of addressing ExAI in NLP independently from other ExAI techniques.

First, some of the explainability methods provide their explanations in terms of specific input features, such as pixels. These features are not as straightforward in textual datasets. The majority of NLP models operate on embeddings that are opaque representations, as opposed to pixels or numerical values. Providing explanations in terms of specific embedding dimensions will not have a practical implication for the explanations. Additionally, the inner workings of deep networks trained on text cannot be easily visualized as opposed to visual networks. Further processing on the inner encodings is required to dissect the learned knowledge.

When a model's decision is explained by the words that contributed to the prediction, further refinement is needed. For instance, explaining decisions in terms of the input features can be easily formulated when the input is numerical or imagery where decisions can be reflected by clear features or pixels. This task becomes challenging with text where the syntactic and semantic features cannot be easily dissected in the input to interdependently serve as explanations. After all, a word is a fusion of syntax, semantics, and previous context. When providing the explanation as a set of words, one can thus inquire if the explanation model is attending to the part-of-speech tag, the entity, the meaning, or the accumulated context.

Long-term dependencies, however, add to the challenges of ExAI methods for NLP models. Namely, textual explanations might not always be a set of consecutive words but words with longer dependencies as opposed to neighboring pixels in images. Finally, multi-lingual support for deep models language models specifically introduces new challenges to explainability where the encoded language semantics and context need further deciphering.

3 RELATED SURVEYS

Given the infancy of the ExAI field, especially in the context of NLP models, there is a handful of surveys that describe its terminology, taxonomy, different methods, and evaluation frameworks.

Doshi et al. introduce the taxonomy of ExAI in Reference [28] while setting the common ground for rigorous evaluation of interpretability of machine learning models through application-grounded, human-grounded, and functionally-grounded settings. Montavon et al. [70] focus on activation maximization techniques, sensitivity analysis, Taylor decomposition, and relevance propagation. Their work is specific to *post hoc* interpretability methods and does not discuss *inherently* transparent models. A subset of the interpretability methods is also surveyed in Reference [85], where a comprehensive study is presented covering activation maximization, network inversion, deconvolutional neural networks, and network dissection-based visualization applied on imagery datasets. Similarly, Guidotti et al. [39] focus on decision rules, features importance, saliency masks, sensitivity analysis, partial dependence plot, prototype selection, and neurons activation methods while mainly studying image and tabular datasets.

In Reference [37], a brief survey that discusses linear proxy models, decision trees, automatic rules, and saliency maps is presented. This survey studies a few approaches that are *inherently* explainable such as attention networks and disentangled representations that are designed to generate explanations. Later, Nguyen et al. [74] discuss how the activation maximization approaches evolved. Their survey is limited to imagery datasets and is very specific to optimization techniques in activation maximization. The survey of Reference [110] is also specific to explainability methods in the medical field, which implies signals and imagery datasets. Very recently, Reference [23] introduced the opportunities and challenges of ExAI while discussing the limitations of certain methods, and Reference [84] presented ExAI methods applied to reinforcement learning settings.

The majority of the surveys in the literature are either brief [29, 37] or are focused on imagery datasets [23, 74, 85]. It is not until late 2020 that a survey on ExAI integrated with NLP models was introduced in Reference [22]. In this survey, Danilevsky et al. categorized the explanation types and methods while focusing on visualizing techniques and presenting some gaps and research direction in the ExAI-NLP area. A few months later, a survey on explanation-based human debugging of NLP models [56] has been released. The survey targets explanatory debugging and human-in-the-loop debugging systems.

While both papers, References [22, 56], targeted a similar framework, there are clear differences among the scopes, depth, and breadth. Our work is the first to survey existing work on the explainability of word embeddings and the attention mechanism. Moreover, Reference [22] focuses on the interpretability methods that explain individual predictions, and Reference [56] focuses on interactive human-in-the-loop learning, whereas our work is broader. We instead report on literature that interprets the knowledge encoded by language models. Finally, Reference [22] claimed that only four papers are found in the literature targeting global explanations that are defined as the study of the predictive process independently of any particular input. While we acknowledge the difficulty of finding such papers due to proper tagging with ExAI keywords, we dedicate three sections (5–7) to survey over 50 papers that are missed by Reference [22], and we highlight attempts to understand how NLP models process inputs and the information they encode.

Figure 1 visualizes existing surveys since 2017 and highlights the broadness of their scope, the data types that they study, their impact on the ExAI community (reflected by their citation number), and whether evaluation methods are included.

4 TERMINOLOGY

In this survey, we study interpretability methods on NLP models over the *how*, the *what*, and the *which* dimensions. Our first dimension is specific to *how* the models are being explained, i.e., by design or in a *post hoc* manner. The former methods develop models from scratch, such that they are *inherently* interpretable while the latter methods explain pretrained black-box models. Although interpretable models are *inherently* transparent, they require modifying the model's

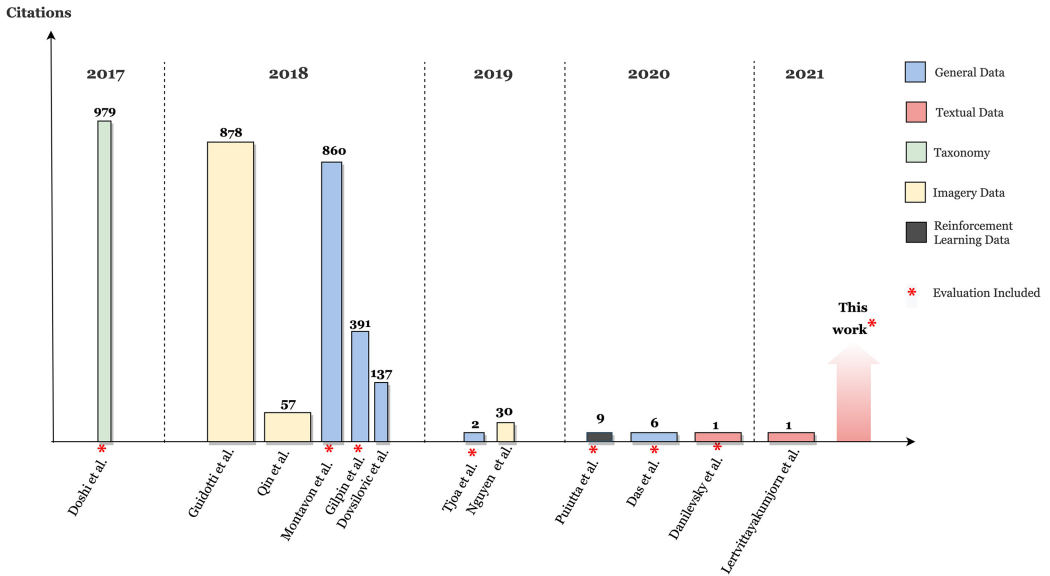


Fig. 1. Summary of surveys on ExAI aggregated by year, citation number, evaluation, and type of considered data. The thickness of the bar reflects the scope of the presented work categorized into two groups: *broad* and *narrow*. The thick bars represent broad scopes and the thin bars the narrower ones.

architecture and retraining huge models. A recent study [96] outlines the difference between both frameworks and highlights the areas where developing explainability methods for black-box models should be avoided in high-stake decision-making environments. In the second dimension, we categorize the interpretability attempts into three categories according to *what* they explain. The first category interprets the word embeddings, which is on the input level of most of the deep NLP networks. The second category interprets the inner representations of RNNs and transformers, which is on the processing level of the DL. Finally, the third category interprets individual model decisions with respect to specific input features or neuron activations that are on the output level of the networks. The third dimension addresses the question of *which* models are being interpreted by making the distinction between explanation methods that are model agnostic, i.e., can operate on any machine learning model, and those that are model specific, i.e., tailored for specific architectures. Figure 2 highlights the contributions of this article in terms of these three dimensions within the terminology used in the ExAI framework and highlights the difference between our work and Reference [22].

5 INTERPRETING WORD EMBEDDINGS

In this section, we discuss the explainability of word embeddings that constitute the input to the majority of NLP models. Word embedding models are the result of various optimization methods and deep structures that represent words as low-dimensional dense continuous vectors [67, 79]. Although effective at encoding semantic and syntactic information, those vectors are notoriously hard to interpret. Additionally, they show little resemblance to the theories of lexical semantics, which are based on familiar, discrete classes and relations.

Interpretability of word embeddings is not only essential for developing transparent models but also is desirable for lexicon induction, efficiency, and fairness [31]. For instance, if interpretable,

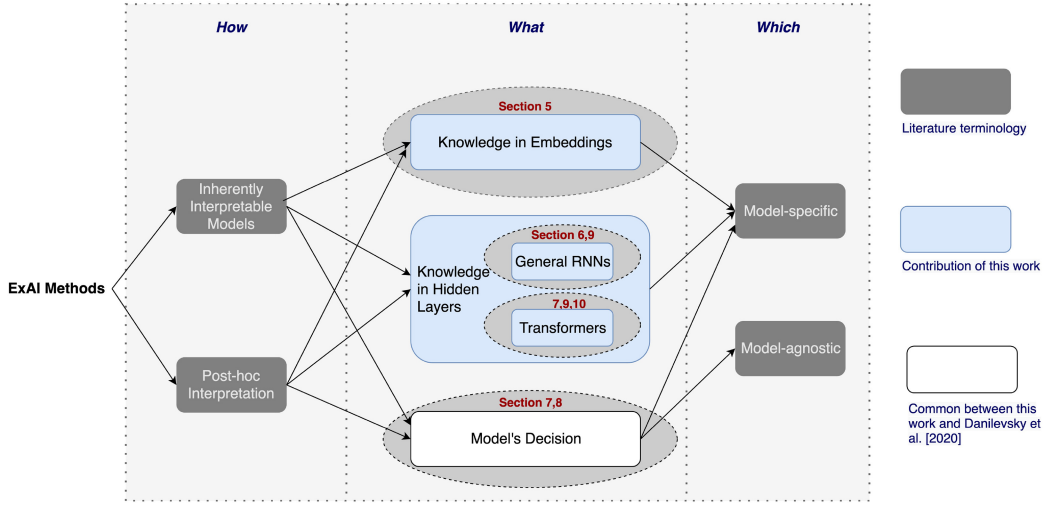


Fig. 2. The dimensions studied in this work within terminology used in the ExAI framework and a visual comparison to the work of Reference [22]. Missing arrows between consecutive dimensions reflect the infeasibility of the integration of the proposed dimensions.

then semantically or syntactically similar words can be easily extracted in lexicon induction. Additionally, the evaluation embeddings would be transparent by examining the information encoded by the individual vectors. Interpretability has also a computational advantage on many classification tasks where irrelevant dimensions can be disregarded. Finally, *fairness* can be ensured by removing dimensions encoding protected attributes such as gender or race [12].

Recently, different approaches have been suggested to improve the interpretability of word embeddings. Such methods, discussed next, rely on visualization [65], impose sparsity [34, 73, 105] and dimension-specific constraints [128], use rotation techniques [32, 76, 95], or incorporate external knowledge [33, 49] and contextual information [80, 81, 108] to derive more interpretable embeddings. Other approaches rely on neighborhood analysis to quantify interpretable characteristics of a given embedding model [92]. Such analysis allows us to dissect the information that a given word embedding encodes and to explain the correlation between the embedding model on different tasks. All these approaches are *model specific* and the distinction between *post hoc* and *inherent* interpretability is later highlighted in Table 2.

5.1 Sparsification of Embedding Spaces

Dense word embeddings cannot solely provide meaningful representations. For instance, in dense word embeddings, the representation of a word w is a dense vector of small positive or negative values spreading several hundred dimensions. Those dimensions are also active (i.e., having non-zero values) for words of different types and domains. Consequently, an active dimension in a dense word embedding model cannot imply a specific semantic or syntactic form. Sparsifying word embedding models can thus map dimensions to meaning or syntax, making embeddings more explainable. To illustrate this concept, we visualize in Figure 3 five words for five randomly chosen dimensions from dense (SVD300) and sparse (NNSE1000) embedding models as derived by Reference [73]. We observe that the words in each dimension are more semantically coherent for the sparse embedding model and thereby more interpretable.

Murphy et al. [73] used a matrix factorization algorithm, **Non-Negative Sparse Encoding (NNSE)**, to derive the first distributional model that satisfies sparsity, effectiveness, and

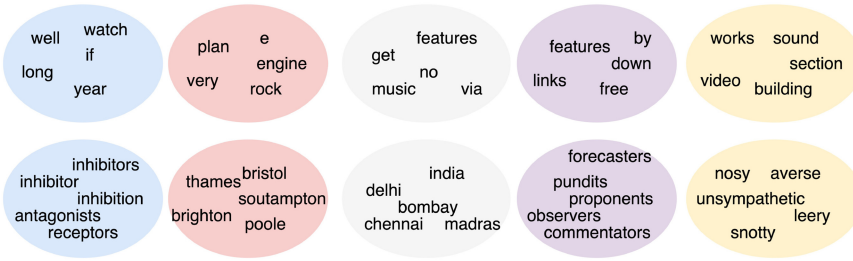


Fig. 3. Sets of five words for five randomly chosen dimensions from a dense (SVD300) and sparse (NNSE1000) embedding model as in Reference [73]. Better semantic coherence is observed in the sparse NNSE1000 model.

interpretability. The authors decompose the input representation $X \in \mathbb{R}^{m \times n}$ into two matrices, $A \in \mathbb{R}^{m \times k}$ and $D \in \mathbb{R}^{k \times n}$, subject to sparsity and non-negativity constraints. Their problem is to find A, D that minimize $\sum_{i=1}^m \|X_{i,:} - A_{i,:} \times D\|^2 + \lambda \|A_{i,:}\|_1$, subject to $A_{i,j} \geq 0$ and $D_{i,:} D_{i,:}^T \leq 1$.

The optimization in the NNSE method requires heavy memory usage and cannot effectively deal with streaming text data. Consequently, an **online approach to deriving an interpretable word embedding model (OIWE)** is proposed in Reference [64]. OIWE uses projected gradient descent to apply non-negative constraints on non-negative methods such as Skip-Gram. An unconstrained optimization problem is later proposed in Reference [34] to transform any distributed representation into sparse or binary vectors. Using sparse coding, each input vector X_i is represented as a sparse linear combination of basis vectors. Using online adaptive gradient descent, longer and sparser vectors are derived as more interpretable “overcomplete” representations.

Later, in Reference [104], the authors employed de-noising k -sparse auto-encoder to obtain an interpretable transformation of input embeddings, **SParse Interpretable Neural Embeddings (SPINE)**. Input embeddings are projected into a new space $\in \mathbb{R}^{m \times v}$ where embeddings are both sparse and non-negative. De-noising k -sparse auto-encoder is used to train the model by minimizing the combination of the reconstruction loss, the average sparsity loss, and the partial sparsity loss over the dataset while capturing the sparsity constraints.

Deriving the embedding models, discussed so far, relies on optimization methods that mainly operate in ϵ -accurate regimes. Operating in a *post hoc* manner might cause an accumulation of inaccuracies. Some methods introduce sparsity as a post-processing step. For instance, Sun et al. [105] directly apply the sparsity constraint while computing the word embeddings of *Word2Vec*. This is achieved by introducing the l_1 regularizer and employing regularized dual averaging to produce sparse representations. Additionally, Panigrahi et al. [75] developed an unsupervised method to generate *Word2Sense* where each dimension encodes a fine-grained *sense*. *Word2Sense* embeddings are probability distributions over *senses* where the probabilities of each word and sense are estimated and the Jensen Shannon divergence [35] is then used to compute word similarities to eliminate redundant senses.

5.2 Rotation of Embedding Spaces

From a linear algebra perspective, having a *transparent* basis, every embedding vector can be explained as a combination of understandable concepts. To identify these interpretable dimensions (i.e., basis), word spaces can be *rotated* while preserving the encoded information.

Rothe et al. [95] formulated this rotation as a decomposition of an embedding space into two components: an interpretable orthogonal subspace and a “remainder” subspace. The resulting subspaces and their orthogonal complements form the basis for an embedding *calculus* that supports certain operations. The goal becomes to find an orthogonal matrix that transforms the embedding

space into an interpretable one with fewer dimensions. Different lexicons such as opinion, POS, and emotion are used to train the transformation matrix to minimize the distance between lexicon pairs and words with identical labels. Consequently, the authors extended the “king - man + woman = queen” analogy into operations like “-1× hate = love.”

In Reference [76], interpretability is induced by factor rotation that reforms the word embedding matrix to have a simple structure by a linear transformation. The rotation encourages each row and column (word vector and dimension, respectively) to have a few large values. More specifically, the rotation is a post-processing step that computes a rotated matrix minimizing the rotation criterion. The latter was introduced in Reference [20] by forcing a low complexity on the rows and the columns of the rotated matrix and is minimized using the gradient projection algorithm. This approach can be seen as a combination of *sparsification* and *rotation* where embedding models are rotated while encouraging low complexities in the values. Likewise, Reference [32] computes orthogonal transformations by randomly sampling word vectors and their target before deriving the orthogonal Procrustes closed-form solution.

The approaches discussed above enhance the explainability of existing model embeddings. However, the derived explanations are simple linear algebraic analogies, i.e., at the level of simple grammatical relations such as negation, grammatical gender, and prefix derivation. In Reference [1], the explainability of embeddings was addressed from a different angle. Instead of making the embedding models more explainable, Allen et al. [1] targeted well-known linear relationships and derived their probabilistically grounded interpretations. Specifically, the authors aim at explaining why the embeddings often satisfy $w_{b*} = w_a * -w_a + w_b$ when word embeddings are trained using only word co-occurrence. This is achieved by first defining *paraphrasing* as “an equivalence drawn between words and word sets by reference to the distributions they induce over words around them.” Then, the authors show that *paraphrasing* determines linear relationships hold whenever the embeddings factorize pointwise mutual information. Finally, the linear relationship follows when analogies between words are interpreted as word transformations sharing the parameters. The mathematical definitions and proofs provided in the article establish the first rigorous explanation of the embeddings’ linear relationships.

5.3 Integrating External Knowledge

The previous two methods discussed the interpretability of word embeddings without the incorporation of external knowledge. However, existing lexicons and ontologies help in building embeddings spaces that can, by construction, reflect the semantic and syntactic relationships and that are thus more interpretable. For instance, Faruqui et al. [33] enhance the interpretability of embeddings by fine-tuning the embedding model using relational information from semantic lexicons. This is achieved by proposing a graph-based learning framework, “retrofitting,” for incorporating lexical relational resources. Given an un-directed graph $\Omega = (V, E)$ of semantic relationships and a word embedding model $\hat{Q} = (\hat{q}_1, \dots, \hat{q}_n)$, the goal is to learn a *refined* embedding model, $Q = (q_1, \dots, q_n)$, such that its column vectors (word embeddings) are close in distance to their counterparts in \hat{Q} and to their adjacents under Ω . The optimization problem is thus to minimize $\sum_{i=1}^n [\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2]$. The embeddings are first trained independently of the semantic lexicons and then through “retrofitting,” Q is computed by solving a system of linear equations using an efficient iterative updating method.

In some particular domains, knowledge graphs go beyond syntax and semantics and develop knowledge basis on higher-level categorical and scientific relations. The medical domain is a perfect illustration where Reference [49], for instance, incorporates the rich categorical and taxonomic knowledge in the biomedical domain to leverage the interpretability of medical

embeddings. The authors learn a transformation matrix that transforms the word embeddings to a new interpretable space according to the biomedical taxonomy while retaining its expressive features. Similarly, Pelevina et al. [77] utilize an *ego-network* to transform word embeddings into sense vectors. The authors define an *ego-network* as a set single nodes, *ego*, along with the nodes they are connected to, *alter*. After learning word embeddings and building a graph of nearest neighbors based on vector similarities, word senses are induced by clustering the *ego-network*. Such word senses can be effectively used in word sense disambiguation.

Although based on a sound mathematical formulation, the previously discussed approaches do not explicitly show the practicality of the derived explainable embeddings in tasks such as *de-biasing* the embedding models. Instead, Reference [128] addressed the gender-neutral word embeddings problem and aimed at eliminating gender influence while preserving essential gender information in specific dimensions. This was done by categorizing all the words into male-definition, female-definition, and gender-neutral according to WordNet definitions. Then, embeddings were learned by minimizing using stochastic gradient descent a combination of (1) the word proximities, (2) the negative distance between words in the female and male-definition seed words, and (3) the difference between female words and their male counterpart.

5.4 Contextualized Embeddings

The word embeddings that we discussed so far are functions where every word, regardless of whether it has more than one meaning, has a unique embedding vector. Peters et al. [81], proposed an embedding model, where words are represented by contextualized vectors that model the syntactic and semantic characteristics of a word as well as disambiguation across linguistic contexts, i.e., in the case of *polysemy*. Disambiguation is possible due to the formulation of the word representation as a function of the entire sentence in pre-trained bidirectional language models. While such models are discussed in Section 7, we focus here on the explainability of their embedding layer rather than the full model.

The work of Reference [80] empirically “dissects” the contextualized embeddings by evaluating their performance in a suite of four NLP tasks. The authors investigate the intrinsic properties of contextual vectors that are independent of the NLP model and architecture details by studying how semantic and syntactic information is modeled throughout the network’s depth. The results show that morphology is encoded in the word embedding layer, the local syntax in the early contextual layers, and semantic information in the deeper layers.

Tenney et al. [109] studied the way sentence structure is modeled across a range of syntactic, semantic, local, and long-range phenomena in contextualized embeddings. Through edge probing, the authors showed that such embeddings outperform their non-contextualized counterparts on syntactic tasks and not on semantic tasks, which shows that they are better at encoding syntax than higher-level semantics. Moreover, contextualized representations are able to encode long-term linguistic information that helps disambiguate longer-range syntactic dependencies relations and structures. Contextualized embeddings are also studied within the gender bias framework in [127]. The analysis showed that some training data contain significantly more male than female entities, which gets reflected in embeddings that systematically encode gender information in an unequal manner.

5.5 Evaluating Embeddings Interpretability

The research work described above attempts at improving the interpretability of word embeddings. The question that can be immediately asked targets the evaluation of the interpretability of the obtained embedding spaces. While a common rigorous evaluation method, that is based on

Table 1. Precision on the Word Intrusion Task on the State-of-the-art Sparse Embedding Spaces

NNSE [73]	SPOWV [34]	SPINE [104]	Word2Sense [75]
92.33	41.75	74.83	75.3

well-designed bench-marking datasets, is yet to be developed, researchers are currently utilizing a set of interesting experiments summarized in this section.

The interpretability of the sparsification methods is commonly evaluated through word intrusion, which seeks to quantify how coherent the dimensions of a learned word representation are. The experiment goes as follows: From the learned representation, a dimension i is chosen and the vocabulary words are then ranked according to the variance of their values. The four highest-ranked words are then chosen across each dimension and a word from the bottom half of the list is then added as an “intruder” that human judges are asked to identify. The precision of the human judges across different state-of-the-art sparse embeddings is reported in Table 1. In Reference [105], the word intrusion task is extended and a new evaluation metric that does not require any human assessment. The intruder word should be different from the top four words which should be similar. Consequently, the ratio of the distance between the intruder and top words to the distance between the top words themselves is used as a quantitative metric. The higher the ratio, the more interpretable the embedding state.

Additionally, Rothe et al. [95] used the cosine similarity to decide whether two words are synonyms or antonyms, which reflects the level of interpretability of word connotations. The cosine similarity between embeddings was also used in polarity spectrum creation where the main task is to predict the spectrum of a word in a certain query. The morphological analogy was also used to test how well these embeddings can encode the POS tags in the computed subspaces. These experiments test the quality of a subspace rather than its interpretability.

The work in Reference [33] evaluates the interpretability of embeddings as their ability in capturing semantic and syntactic aspects. Such tasks include the word similarity and the synonym-selection tests that are aligned with Reference [95] discussed above. The syntactic relations test evaluates how well the embeddings can encode relations of the form “ a is to b as c is to d .” Sentiment analysis tests the knowledge encoded in these representations. While these tests better target the interpretability of the representations in terms of their semantic and syntactic relations, they do not explicitly reflect the interpretability of certain dimensions.

Peters et al. [81] dissect the word representations in bi-directional models and evaluate their interpretability in terms of the quality of the interpretation on tasks such as semantic role labeling, constituency parsing, and named entity recognition. Later, this set of experiments is extended in Reference [80] to include question answering, sentiment analysis, and textual entailment.

5.6 Discussion

The evaluation methods discussed above mainly reflect the quality of embeddings rather than their interpretability. For instance, evaluating the embeddings on a POS tagging task shows how well they encode some syntax properties without reflecting the exact dimension, or the set of dimensions, where such properties are encoded. In other words, the majority of experiments report an overall performance on some NLP tasks without addressing the alignment between a dimension and a particular context or syntactic aspect.

Some interesting experiments are reported in References [34, 73, 75, 104]. Murphy et al. [73] qualitatively assess the interpretability of dimensions of the NNSE embeddings by investigating the dominating dimensions in the NNSE representations. Similarly, Faruqui et al. [34] consider the

Table 2. Summary of Existing Work on the Interpretability of Word Embeddings

Method	Needs Existing Embedding?	Common Evaluation	Posthoc / Inherently	References
Sparsification	✓	✓	<i>Inherently</i>	[34, 64, 73, 104]
	x	✓	<i>Inherently</i>	[75, 105]
Rotation	✓	x	<i>Inherently</i>	[1, 32, 76, 95]
Integrating External Knowledge	✓	x	<i>Inherently</i>	[33, 49, 77, 128]
Contextualized Embeddings	✓	x	<i>Post hoc</i>	[80, 109, 127]
			<i>Inherently</i>	[81]

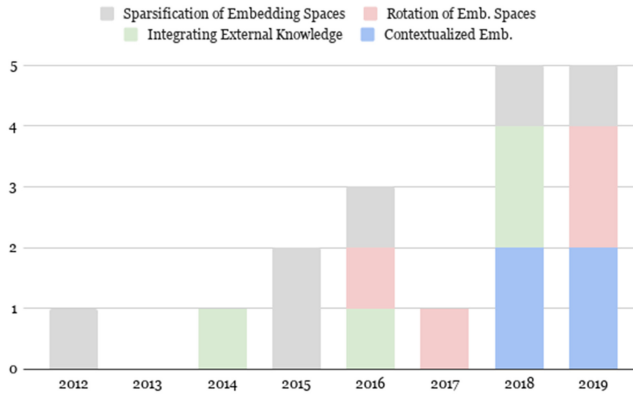


Fig. 4. Number of the surveyed papers in each method reflecting the interest since 2012.

dimension interpretability and qualitatively assessed whether the top-ranking words for a particular dimension display semantic or syntactic groupings. Reference [104] extend this experiment to SPINE and qualitatively compare the interpretability of SPINE to that of SPOW with two baseline embedding models, *word2vec* and *Glove*. The authors examine the top participating dimensions for some sampled words and study the top words from the participating dimensions. Similar experiments are conducted by Reference [75] in *word2sense*.

These experiments are paving the way toward a rigorous evaluation of embeddings' explainability but are still lacking the commonality and objectivity aspects. Finally, the majority of these methods operate on embedding models and are thus *model specific*. However, some of the obtained models are *inherently* interpretable, whereas others need further processing. Table 2 summarizes the discussed methods within the dimensions discussed in this work while highlighting their reliance on common evaluation schemes and whether they need existing embedding spaces. Figure 4 reflects the interest in the interpretability of word embeddings by showing the number of papers we surveyed within each category since 2012. One can see that the sparsification of embeddings was one of the earliest methods in the field. After the success of bidirectional transformer models in 2017, contextualized embeddings are attracting researchers to study their interpretability. Interest in other methods seems to be consistent and a general interest in the interpretability of word embeddings seems to be increasing.

6 WHAT DO RNNs LEARN?

After discussing the interpretability on the input level, we focus next on the processing level. Hence, we dissect the inner representations of RNNs. These methods either implement *inherently*

interpretable RNN architectures or try to interpret existing RNN architectures in a *post hoc* manner. Both approaches, however, operate in a *model-specific* way.

6.1 *Post hoc* Interpretation

Most of the *post hoc* interpretation methods try to dissect hidden knowledge in trained deep networks from a syntax and semantic lens. One such lens is *compositionality*, defined in Reference [57] as a way to understand how such networks build sentence semantics from individual words. The authors develop strategies based on heatmap and t-sne visualization by plotting unit values to visualize negation, intensification, and concessive clauses. Similarly to Reference [51], but using the unit's salience, the authors are able to compute the amount that each input contributes to the semantics of a sentence. These strategies show that **Long Short-Term Memory Networks (LSTMs)** are able to perform well due to their ability to maintain a sharp focus on essential keywords. Moreover, sentiment analysis tasks are sensitive to some dimensions in an embedding. More importantly, neural models in NLP are able to learn the properties of local *compositionality* in a subtle way while respecting asymmetry as in negation.

Focusing on syntax, Blevins et al. [11] develop a set of experiments that investigate how the internal representations in deep RNNs capture soft, hierarchical syntactic information without explicit supervision. First, the authors extract word-level representations produced by layers of the RNN trained on tasks such as dependency parsing, machine translation, and language modeling. A classifier is then trained to predict the POS tag and the parent, grandparent, and great-grandparent constituent labels of that word. Another classifier is trained to check whether a dependency arc exists between two words. The evaluation of these classifiers shows that the representations learned by RNNs encode syntax beyond the explicit information encountered during training. The results demonstrate that the word representations produced in RNNs at different depths are highly correlated with syntactic features. In particular, deeper layers are shown to capture higher-level syntax notions.

These findings agree with those of Reference [111], where recurrent architectures are compared to non-recurrent ones for their ability to model hierarchical syntax. Experiments on subject-verb agreement and logical inference show that recurrent architectures, LSTM specifically, are notably more robust and present better generalization guarantees when longer sequences are encountered. The same quest is applied to LSTMs in Reference [60] to study how syntactic structures are encoded. Through number agreement in English subject-verb dependencies experiments, the authors show that LSTMs are able to capture a considerable amount of grammatical structures, but more expressive architectures may be required to reduce errors in particularly complex sentences. Number agreement is also studied later in Reference [40] by including nonsensical sentences where RNNs cannot rely on semantic or lexical cues and by comparing to human intuition to show that long-distance agreements are reliably encoded in an RNN. In Reference [53], the role of context in LSTMs is analyzed through ablation studies by shuffling, replacing, and dropping prior context words. LSTMs were found to be able to use about 200 tokens of context on average. However, LSTMs struggle to distinguish between nearby context from distant history. More interestingly, the model pays attention to the order of words only in the recently processed sentence. LSTMs' predictions have been empirically explored in Reference [52] along with the LSTMs' learned representations. Specifically, long-range dependencies are investigated in character-level language to seek cells that identify high-level patterns such as line lengths, brackets, and quotes. Karpathy et al. [52] study *n*-gram model to conclude that LSTMs' performance improves on characters that require long-range reasoning.

The discussed approaches address particular semantic and syntactic structures and study *where* and *how* these structures are encoded. Nonetheless, these approaches do not study how the

input is processed through these inner encodings. Very recently, Hou et al. [45] suggested a novel interpretation framework inspired by computation theory. The authors are the first to draw the analogy between **Finite State Automata (FSA)** and an interpretable inner mechanism for sequential data processing. In their work, FSAs are learned from trained RNN models to achieve better interpretability of the latter. When inputting several sequences to an RNN model, the hidden state points are gathered, and similar ones are clustered. The FSA states are the clusters, and the transitions between the states arise when one item of the input sequence is processed. Aside from the novel connection with FSA, an interesting advantage of Reference [45] is its ability to provide global explanations in classification tasks.

6.2 Inherently Interpretable Models

Inherently interpretable RNNs are trained in an explainable way by adding transparency constraints [118] and exploring tree and graphlike structures [59, 106]. In Reference [118], the authors develop an *inherently* interpretable RNN, SISTA-RNN, based on the sequential iterative soft-thresholding algorithm and the idea of *deep unfolding* [43]. By allowing the RNN parameters to be seen as the parameters of a probabilistic model, the weights and outputs can retain their meaning. Traditionally, given input-output training pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i=1, \dots, N}$ and model parameters θ , RNNs learn θ according to the following optimization problem: $\min_{\theta} \sum_i f(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ subject to $\hat{\mathbf{y}}_i = g_{\theta}(\mathbf{x}_i), i = 1, \dots, N$, where f is the loss function and g is the conventional black-box RNN. By changing the optimization constraint to $\hat{\mathbf{y}}_i = h_{\theta}(\mathbf{x}_i), i = 1, \dots, N$ and $h_{\theta}(\mathbf{x}_i) = \arg \min_z \mathcal{P}_{\theta}(z, \mathbf{x}_i)$ with h attempting to solve an optimization problem \mathcal{P}_{θ} that corresponds to a principled probabilistic model, the parameters θ become interpretable. SISTA-RNN is proved to achieve a higher performance on a sequential compressive sensing task.

Tai et al. [106] improved the semantic representation of LSTMs by building a semantic tree on semantic topologies. Tree-LSTMs are similar to the traditional LSTM units but differ in the gating vector and memory cell update that are dependent on the state of the children (tree notion). A forget gate per child allows Tree-LSTM to select the information in each child. Later, Liang et al. [59] learn interpretable representation following a wise graph construction paradigm. Their *structure-evolving* LSTM first considers each data element as a separate node in a graph. Then, nodes with high compatibility are recursively merged to form richer and more interpretable encodings. Similarly, graph LSTMs [78] are formulated to incorporate linguistic analysis. This approach is shown to encode richer linguistic knowledge improving the performance of relation extraction [41, 78, 126, 129]. Although both LSTMs outperformed existing systems, their evaluation remains under-investigated. For instance, Reference [106] evaluate their LSTMs on sentiment analysis and semantic relatedness tasks. Moreover, the effectiveness of *structure-evolving* LSTMs is mainly evaluated on a non-NLP-related image segmentation task without thorough testing on other modalities.

Neural module networks jointly train deep “modules” for visual question answering offering visual explanations for an answer [4, 46]. While such networks are not fully within a textual modality, i.e., they accept the question as text and an image as a reference, we briefly discuss them in this survey for completeness purposes. Neural module networks learn to parse questions as executable modules to understand well synthetic visual QA domains [42]. The questions are analyzed in a semantic parser before determining the basic computational units that contribute to the answer [4]. Andreas et al. [3] learn the parameters for these modules jointly via reinforcement learning on (world, question, answer) triples. Gupta et al. [42] extend this approach and introduce modules that reason over a paragraph of text through symbolic reasoning over numbers and dates. Furthermore, an unsupervised auxiliary loss is suggested to extract arguments explaining specific events in the text.

Table 3. Summary of Interpretability Attempts at Understanding the Inner Representations of RNNs

Reference	<i>Inherently/ Post hoc</i>	Model	Method	Evaluation Method
[52] (2015)	<i>Post hoc</i>	LSTM	Identification of high-level patterns	Reasoning on n -gram models, Subject-verb agreement
[57] (2016)	<i>Post hoc</i>	LSTM	Heatmap	T-SNE visualization
[60] (2016)	<i>Post hoc</i>	LSTM	Probing	Subject-verb agreement and logical inference
[11] (2018)	<i>Post hoc</i>	RNN	Correspondence between network depth and syntactic depth	POS tag and dependency classification
[40] (2018)	<i>Post hoc</i>	RNN	Comparison to human intuition	Subject-verb agreement
[53] (2018)	<i>Post hoc</i>	LSTM	Ablation	Context localization
[53] (2018)	<i>Post hoc</i>	LSTM	Ablation	Context localization
[106] (2015)	<i>Inherently</i>	LSTM	Integration of semantic topologies	Sentiment analysis and relation classification
[118] (2016)	<i>Inherently</i>	RNN	Sequential iterative soft-thresholding algorithm and deep unfolding	Sequential compressive sensing task
[59] (2017)	<i>Inherently</i>	LSTM	Integration of knowledge graphs	Semantic object parsing
[4] (2016)	<i>Inherently</i>	Module Networks	Reinforcement Learning	Visual question answering
[46] (2017)	<i>Inherently</i>	Module Networks	Joint learning	Visual question answering
[42] (2019)	<i>Inherently</i>	Module Networks	Unsupervised auxiliary loss	Reasoning over text
[41, 78, 126, 129] (2017–2020)	<i>Inherently</i>	LSTM	Encoding of richer linguistic knowledge	Relation Extraction

6.3 Discussion

Table 3 summarizes the aforementioned approaches while highlighting their interpretability and evaluation methods. While Reference [57] is inspired by a non-text-specific interpretability approach, the rest of the methods are designed to suit NLP models. One can clearly see that a common evaluation is not established yet, and some methods can be subjective [40] while others rely on human-annotated data [11, 106]. Such annotations are not directly related to explainability; they, however, reflect some semantic or syntactic aspects such as subject-verb agreement or POS tags. Although a good performance can imply that the specific semantic or syntactic aspect is indeed encoded in the model, it cannot directly imply explainability. Localization of the encoded aspect in the model as well as input analysis in terms of these aspects are further needed to achieve pure model transparency.

7 WHAT DO TRANSFORMERS LEARN?

After discussing the interpretability of general RNNs, we consider the popular transformer models [114]. Such models are based on attention mechanisms to handle ordered sequences of data. Transformers follow the encoder-decoder structure using stacked multi-head self-attention and

Table 4. Summary of Transformer Visualization Methods

Reference	Method	Model
[54] (2017)	Visualization of attention weights	Neural machine translation models
[62] (2018)	Perturbation-driven	Natural language inference and machine comprehension models
[101] (2018)	Visualization of encoder, decoder, attention, prediction, and beam search	Sequence-to-sequence models
[44] (2019)	Linguistic annotations, masking, and nearest-neighbor search	BERT
[115] (2019)	Visualization on the level of attention-head, model, neuron	BERT and GPT-2

fully connected layers. **Bidirectional Encoder Representations from Transformers (BERT)** [25] and GPT-2 [86] have been trained on huge text corpora and are currently used as state-of-the-art NLP models after fine-tuning on specific tasks. Different approaches have been studied to understand the inner dynamics of transformer models and to visualize the attention weights to better understand *how* they process input and *why* they do it so well. All the discussed approaches are *model specific* and operate in a *post hoc* manner.

7.1 Visualization of Transformers

The Explainability of transformers has been extensively addressed from a visualization perspective by developing tools that allow the user to interact with such models to understand their inner mechanisms. For instance, the work of Reference [54] presents an interactive tool to visualize attention and provides an interface to dynamically adjust the search tree and attention weights. Similarly, in Reference [62], the authors proposed a flexible visualization library to visually analyze models for natural language inference and machine comprehension that relies on a perturbation-driven exploration strategy. Also, in Reference [101], SEQ2SEQ-VIS, visual analytics is presented for sequence-to-sequence model debugging by visualizing the five stages of a seq2seq model: encoder, decoder, attention, prediction, and beam search. SEQ2SEQ-VIS also describes the knowledge that the model has learned by relying on transitions of latent states and their related neighbors. Finally, SEQ2SEQ-VIS provides an interactive method to manipulate the model internally and observe the impact on the output.

ExBERT [44] is another interactive visualization tool that uses linguistic annotations, masking, and nearest-neighbor search to provide insights into contextual representations learned in transformer models in three main components: (1) the attention, (2) the corpus, and (3) the summary view. Similarly, Vig [115] presents a tool to visualize BERT and GPT-2 models at three different granularity levels: the attention-head level, the model level, and the neuron level. Table 4 summarizes these methods and the specific models on which they are applied.

7.2 Is the Attention Mechanism *Inherently* Interpretable?

Controversy has accompanied attention mechanisms since their introduction. While some attention weights can provide reliable explanations [72]; some researchers showed that attention distributions are not easily interpretable and require further processing [13, 47].

In an attempt to investigate these controversies, Vashishth et al. [112] manually analyzed attention mechanisms on several NLP tasks. The experiments showed that attention weights are interpretable indeed and are correlated with feature importance measures capturing several linguistic notions. Similar findings were obtained in Reference [72] on medical tasks where attention weights were efficient in the selection of most relevant segments in a medical document.

These methods, however, are specific to particular domains and linguistic notions and might not be easily extendable to higher-level knowledge structures. Hence, a parallel line of work attempted at proving that “attention is not explanation” by considering more general correlations experiments. For instance, Reference [47] show that there is no frequent correlation between the attention weights and feature importance methods. Moreover, the authors identify attention distributions that yield equivalent predictions through correlation computation.

Prior to these researches, Reference [91] focused on the reasoning capabilities of transformers by considering the recognizing textual entailment task. For this purpose, the attention patterns are visualized on hand-picked validation samples. Word-by-word attention is shown to resolve synonym relationships, match multi-word expressions to single words, and ignore irrelevant parts. Moreover, when a deeper semantics level or common-sense knowledge connects two sentences, attention seems to be capable of capturing the semantics. However, when two sentences are not related, attention seems to be dominated by the last output vector.

The authors of Reference [116] study the interaction between the syntax and the attention weights in GPT-2 [86]. The authors investigate the alignment between syntactic dependency and attention weights. The authors aggregate over the corpus the percentage of total attention of a given head that attends to tokens belonging to the given POS tag with syntactic features. The same aggregation method is used to test the alignment between attention and dependency by computing the proportion of attention between tokens that are components of dependency relations. To explore how long- and short-distance relations are captured, the number of tokens spanned by each head is computed, then attention dispersion is computed based on entropy. The heatmap shows that most POS tags are disproportionately targeted by one or more heads. Attention heads that focus on one POS tags vary according to layer depths: Determiners are targeted in the early layers, while proper nouns are targeted in deeper layers. Additionally, the alignment between attention and dependency relations is strongest in the middle layers. Heads in the early layers tend to focus on position rather than content, whereas attention heads in deeper layers target specific constructs. Finally, deeper layers capture longer-distance relationships. A moderate correlation is found between the distance and entropy of attention and the attention distance is negatively correlated with dependency alignment.

A common limitation of the discussed approaches is the lack of a unified definition of the explainability of attention. Formal explainability definition in NLP, or the lack thereof, is addressed in Reference [13], where *identifiability* of the attention weights is defined as their ability to be uniquely identified from the attention head’s output. The study of *identifiability* is done on attention weights and token contextualized embeddings and the aggregation of context into hidden tokens. Input tokens are shown to retain their identity, whereas the information identity gradually decreases with depth. Pruthi et al. [83] manipulate the attention weights to whitewash problematic tokens in explanations that affect the model fairness or accountability. This is achieved by diminishing the weight assigned to these impermissible tokens.

Table 5 summarizes the controversy around the attention weights while highlighting the methods and the findings of each work. In summary, attention weights are more *inherently* interpretable than the parameters of general deep networks. However, some aspects of interpretability need further processing and investigation of the attention weights.

7.3 Interpretability of BERT

Bidirectional training is introduced in BERT [25] revolutionizing the training of language models. This section discusses the interpretability methods that study the *why* and the *how* such training works so well. Such dissection aids in understanding BERT [48, 108], its weights [19, 87] and limitations [93]. More interestingly, it improves the performance of query retrieving by

Table 5. Summary of the Literature Discussing If the Attention Weights Are Inherently Interpretable

Reference	Method	Interpretable?	Finding
[72] (2018)	Selection of most relevant segments	✓	Ability of attention mechanism to identify meaningful explanations
[112] (2019)	Analysis of weights on text classification and text generation tasks	✓	High correlation between attention weights and feature importance of linguistic features
[91] (2015)	Weights visualization on recognizing textual entailment	X	Dominance of the last output vector over attention in some cases
[27] (2017)	Layerwise relevance propagation (LRP) [6]	X	Importance of LRP to further interpret the attention weights and the internal workings of transformers
[116] (2019)	Alignment between syntactic dependency and attention through visualization and aggregation	X	(1) Disproportionality between heads targeting POS, (2) capturing of longer-distance relationships by deeper layers, and (3) moderate correlation between distance and entropy of attention
[47] (2019)	Correlation and counterfactuals	X	No frequent correlation between attention weights and gradient-based measures of feature importance
[13] (2019)	Aggregation of context into hidden tokens	X	Preservation of the token <i>identifiability</i> throughout the model and decrease of information identifiability with depth
[83] (2020)	Diminishing attention weights of impermissible tokens	X	Attention-based explanations can be deceived especially within the fairness context

automatically discovering better prompts to use to retrieve and combine more accurate answers as in Reference [50]. The approaches discussed next describe the inner workings of BERT models, their attention weights, and methods used to dissect their inner knowledge.

The work of Reference [87] can be considered as a general evaluation scheme for BERT attention weights. The authors proposed several methods to analyze the linguistic information in BERT. First, heatmaps of attention weights are explored to find linguistic patterns. Second, a maximum spanning tree is constructed for each sentence to check if the syntactic dependencies between tokens have been learned by the network. Sequence labeling tasks are also used to measure how important the learned features are for different tasks. Finally, the encoder weights of a high-resource language pair are used to initialize a low-resource language pair to assess the generality of the learned features. These methods lead to insightful conclusions. Attention is shown to follow four different patterns: paying attention to the word itself, to the adjacent words, and to the end of the sentence. Early layers tend to focus on short dependencies while higher ones focus on long dependencies. As expected, training on larger datasets can induce better syntactic relationships, and syntactic dependencies get to be significantly encoded in at least one attention head in each layer of the model. Moreover, early layers encode syntactic information, whereas semantic information is encoded in the upper layers, and starting from the third layer, information about the input length starts to vanish.

A study more focused on the attention heads is presented in Reference [19]. By means of aggregation, most of the heads are found to put little attention on the current token, whereas there

is a considerable amount of heads that heavily attend to adjacent tokens, especially in the early layers. Moreover, the authors speculate that, when the head's attention function is not applicable, the head attends to the end of the sentence token. Thus, gradient-based measures of feature importance are applied to show that attending to some end-of-sentence token does not have a substantial impact on BERT's output. To study whether attention heads span a few words or attend broadly over many words, the average entropy of each head's attention distribution is computed. The results show that in the early layers, attention heads have widespread attention. To study the alignment between syntactic dependencies and attention weights, attention maps are extracted from BERT. Evaluation of the direction and the prediction of the head shows that certain attention heads specialize in certain dependency relationships. Finally, the coreference resolution test, where the antecedent selection accuracy is computed, shows that BERT heads achieve reasonable co-reference resolution performance.

While References [19, 87] are focused on the attention heads and weights, Tenney et al. [108] develop a layer-based approach to interpret the encoded knowledge in BERT's layers. Tenney et al. study the traditional NLP steps that BERT follows to investigate where linguistic information is formed. The authors employ probing techniques to understand the interactions within BERT better and to study at which layers the BERT network can resolve syntactic and semantic structures. It is worth mentioning that the edge probing technique, introduced in Reference [108], aims at quantifying the degree to which linguistic structures can be extracted from a pre-trained encoder. Two metrics are defined: a scalar mixing weight and a cumulative scoring. The former specifies the most relevant combination of layers when a probing classifier is tested on BERT, whereas the latter quantifies the score improvement on a probing task when a particular layer is considered. The results show that the traditional NLP steps are implicitly followed in order by the network: POS tags are processed earliest, and then constituents, dependencies, semantic roles, and coreference are processed in the deep layers. In accordance with previous work, basic syntactic information is encoded in the early network layers, while higher-level semantic information appears in deeper layers. Moreover, syntactic information is more localizable in a few layers, whereas semantic information is generally spread across the network.

Jawhar et al. [48] address the same question as Reference [108] and attempt at unpacking the elements of language structure learned by the layers of BERT. The authors also use probing techniques to show phrasal representation in the lower layers of BERT is responsible for phrase-level information while richer hierarchy can be found in intermediate layers with surface features at the bottom, syntactic features in the middle, and semantic features at the top. Finally, the deepest layers encode long-distance dependency such as subject-verb agreement. Jawhar et al. [48] also show that similar to treelike structures, the linguistic information is encoded in a classical compositional way.

BERT's ability to encode factual knowledge is studied in References [82, 90] through linguistic tasks such as facts about entities, common sense, and general question answering. BERT is shown to encode relational knowledge without any fine-tuning and without access to structured data like other NLP methods. This suggests that language models trained on huge corpora can serve as an alternative to traditional knowledge bases [82]. Roberts et al. [90] consider fine-tuning language models (such as BERT) to study their ability to encode and retrieve knowledge through natural language queries. The authors found that language models are able to perform well on factual tasks but are expensive to train and more opaque than shallow methods. Moreover, they do not provide any guarantees that knowledge can be updated or removed over the course of training.

Readers are referred to Reference [93] where Rogers et al. survey existing work that dissects how and where BERT encodes semantic, syntactic, and word knowledge. The dissection is done via self-attention heads and throughout BERT layers. Moreover, Reference [93] surveys modifications

to BERT's training objectives and architecture, describe the over-parameterization issue in BERT training, and report some of the compression approaches and future research directions.

7.4 Discussion

Due to their reliance on the attention mechanism, the interpretation of transformers is less challenging than RNNs. In view of the way they are designed, attention weights are relatively more interpretable than the conventional deep networks weights. This design made the evaluation of transformers' inner workings more feasible by visualizing their weights and hidden representations [54, 101, 115]. The controversy that accompanied the attention mechanisms trying to address the extent to which the attention weights are explainable can lead to the following conclusion. Relative to general deep networks weights, attention weights can be thought of as more interpretable. However, solely, their ability to provide full transparency or meaningful explanations is questionable. Further processing is needed to achieve proper transparency of transformer models, especially for long-distance temporal relations. More specifically, when the task at hand is not a simple classification but a more complex task such as translation, question answering, and natural language inference, attention weights might not offer the desired interpretability [27, 47]. Mohankumar et al. [69] argue that when the attention distribution is computed on input representations that are very similar to each other, they cannot provide very meaningful explanations. For this purpose, the authors of Reference [69] diversify the hidden representations over which the distribution are computed for more faithful explanations. Moreover, attention weights become less interpretable in deep layers [13, 116]. Hence, the explanation at deeper layers requires further processing especially that output vectors might dominate the attention weights as in Reference [91].

Moreover, similarly to RNNs (Section 6), the evaluation of interpretability methods on transformers is achieved by designing methods that test specific semantic or syntactic aspect through alignment [116], correlations [47], and general explainability methods such as LRP [6] and weight visualization [91]. The first building block in the common explainability framework of transformers is presented in Reference [13]. However, their definition and metric need to be extended to involve other syntactic and semantic explainability aspects discussed in this section. Finally, developing only *post hoc* interpretation methods on transformers can be explained by the fact that such models are already famous and retraining them is very computationally heavy. Training BERT model on huge corpora, for instance, requires the same energy that five cars consume on average during their lifetime according to Reference [102].

8 EXPLAINING MODEL'S DECISIONS

While the previous section opens the deep *black-box* models to understand their representations, this section focuses on highlighting the evidence supporting their decisions. We categorize these approaches into *post hoc* and *inherently* interpretations.

8.1 Post hoc Interpretation

These methods, consider a pre-trained model and analyze how such a model process a textual input before producing a decision. When the model is black-box the interpretability method is *model agnostic*. If some assumptions are made on the architecture, the method is *model specific*.

8.1.1 Model-agnostic Explanations. Given a black-box model $f : X \mapsto Y$ and an input x , the goal of *model-agnostic* interpretation methods is to explain the individual prediction $f(x)$. The majority of these methods rely on perturbing x according to some distribution D_x .

In Reference [88], Ribeiro et al. present LIME, one of the first state-of-the-art explainability algorithms, which approximates *any* classifier or regressor locally with an interpretable model. LIME presents the interpretation g for the user in terms of comprehensible explanations such as

bag-of-words. Formally, LIME minimizes $\mathcal{L}(f, g, \pi) + \Omega(g)$, where $\mathcal{L}(f, g, \pi)$, a measure of how unfaithful the explanation g is to the original model f with the unfaithfulness is computed in a locality defined by a proximity measure π and $\Omega(g)$, the complexity of the explanation g . The explanation is created by approximating f locally by an interpretable one. $\mathcal{L}(f, g, \pi)$ is approximated by sampling around an input x according to π_x , performing perturbations on the input then explaining linearly, respecting thus local faithfulness.

Later in 2018, Ribeiro et al. [89] argue that the coverage of the explanations generated by LIME is not clear. For instance, “not”, “good” could be an explanation of negative sentiment, and “not”, “bad” could be that of a positive one. Thus, the generated explanation does not clearly state when the word “not” has a positive/negative influence on the sentiment. Ribeiro et al. [89] define “anchors” as if-then rules to generate model-agnostic explanations. A perturbation is applied by replacing “absent” tokens with random words having the same POS tag drawn according to an embedding similarity-based probability distribution. Experiments on visual question answering and textual and image classification show that the anchors model enhances the precision of explanations with less effort to understand and apply.

A landmark in the timeline of the ExAI method is achieved when Lundberg and Lee theoretically showed that many interpretability methods and metrics can be unified in one approach that exploits game theoretical concepts in Reference [63]. Inspired by the Shapley value of game theory, SHAP values are proposed as a measure of the feature importance for a model’s prediction. SHAP is proven to be a unified measure that different methods, such as LIME [88], Deep LIFT [99], and layerwise relevance propagation [6], tried to approximate in the literature. Later, Chen et al. [16] extend SHAP and proposed the L-Shapley and C-Shapley measure by exploiting the underlying graph structure to reduce the number of model evaluations. In Reference [2], explanations of black-box models are formulated as groups of input-output tokens causally related. Explanations are generated by querying with perturbed inputs and solving a partitioning problem to select the relevant components. Variational auto-encoders are used to derive meaningful input perturbations.

We draw the reader’s attention to the fact that the approaches discussed above [2, 63, 88, 89] are not exclusive to textual data. While other similar explainability methods exist [7, 63, 103], they are not validated on NLP tasks and are thus beyond the scope of this work.

8.1.2 Model-specific Explanations. Instead of considering black-box models, Kádár et al. [51] interpret specific recurrent architectures in RNN models by quantifying the contribution of each input to the encoding of a GRU architecture. Their *omission score* is computed by measuring the salience of each word s_i in a sentence $s_{1:n}$ by observing how much the representation of the sentence when omitting the word s_i would deviate. The results show the sensitivity to the information structure of a sentence and selective attention to lexical, semantic, and grammatical relations. Similarly, Arras et al. [5] adapt the Layer-wise Relevance Propagation method of Reference [6] to explain the predictions of accumulators and gated interactions in the LSTM architecture in particular.

References [5, 51] exploit general ExAI methods, such as perturbation and layer propagation, to explain NLP models. Although such methods do not require human annotation or intervention, they might be computationally expensive. Prior to that, researchers developed methods that imitate the thinking process of human beings by relying on human annotations to derive explanations. Those attempts are grouped under the “rationalization” framework that aims at explaining why a specific instance belongs to a category by extracting a “rationale” along with the network annotation. According to Reference [55], a “rationale” can be defined as “sub-sets of the words from the input text that satisfy two key properties. First, the selected words represent short and coherent pieces of text (e.g., phrases) and, second, the selected words must alone suffice for

prediction as a substitute of the original text.” Rationales are proved to help to learn domain-invariant representations that can induce machine attention [9].

In Reference [124], rationale-annotated data are exploited to aid learning by providing the learning algorithm (Support Vector Machines, in this case) with hints as to which features of the input the algorithm should attend. Later, Zhang et al. [125], extended the idea to CNNs by incorporating rationales for text classification in a **Rationale-Augmented CNN (RA-CNN)**. RA-CNN computes a document-level vector representation by summing over its constituent sentence vectors weighted by the likelihood that the sentence is a rationale in support of the most likely class. Adversarial learning is utilized in Reference [123] to improve the performance of rationale extraction so as not to leave any useful information out of the selection. The outcome is also incorporated in Reference [123] into the selection process to improve the predictive accuracy through more comprehensive rationale extraction. Differentiable binary variables are introduced in Reference [10] to further improve the rationale extraction by augmenting the objective with re-parameterized gradient estimates. Game-theoretic concepts are incorporated in Reference [14] to derive a rationalization criterion that approximates finding causal features without highlighting spurious correlations between inputs and outputs.

A common benchmark to evaluate the rationalization attempts is presented in the ERASER framework of Reference [26]. Other approaches include, but are not limited to, contrastive textual explanations [94], representation erasure [58], and information theoretical measures [15].

8.2 Inherently Interpretable Models

While the previous interpretability methods work on pre-trained models, the methods discussed in this section develop models that can classify textual data while explaining their particular decisions. In other words, for a textual input x the models discussed next can output a decision (class) y and an explanation E formulated differently in each reference. Using the terminology of this work, we further cluster these methods in *model-agnostic* and *model-specific* groups.

8.2.1 Model-agnostic Explanations. As an extension to the previously discussed “rationalization” attempts [124, 125], Lei et al. [55] propose an automated approach to extract rationales from the text as subsets of the text words that are coherent and short but lead to the same predictions. For this purpose, the authors train a generator and encoder that learn the classification as well as the explanation. While the former specifies a distribution over subsets of words as candidate rationales, and the latter processes them for prediction. This is achieved by (1) forcing the produced rationale z (set of words) to be sufficient as a replacement for the input text x in predicting the output y , i.e., $\|enc(z, x) - y\|$ small, (2) while maintaining short, i.e., small $\|z\|$, and coherent rationales. Coherency is satisfied by encouraging words to form meaningful phrases (consecutive words) rather than sets of isolated words. Hence, the sum of $|z_t - z_{t-1}|$ is minimized. Then, doubly stochastic gradient descent is used to minimize the following objective over training instances of the form (x, y) :

$$\sum_{(x,y)} \mathbb{E}_{z \sim gen(x)} \left[\|enc(z, x) - y\|_2^2 + \lambda_1 \|z\| + \lambda_2 \sum_t |z_t - z_{t-1}| \right]. \quad (1)$$

8.2.2 Model-specific Explanations. Liu et al. [61] consider local explanations for particular inputs but present a generative explanation framework that learns to generate fine-grained explanations *inherently*, while making classification decisions in a *model-agnostic* manner. The prediction component is composed of a (1) text encoder that takes the input text sequence S and encodes it in a representation vector v_e and (2) a category predictor P that outputs the category corresponding to v_e along with its probability distribution. The explanation component consists of an explanation

generator that takes v_e and generates fine-grained explanations e_c , as a set of words that explain the model's decision. Fine-grained explanations can be thought of as attribute-specific rationales. In other words, in sentiment analysis applications, if a product has three attributes (quality, practicality, and price), then the review "the product has good quality with a low price" can be explained by "low" as a fine-grained explanation for the price and "high" as a fine-grained explanation for the quality. In Reference [61], those explanations e_c are provided in two datasets collected by the authors of the work.

Training consists of minimizing a combination of two-loss factors: the classification loss and the explanation loss. To avoid cases where the generative explanations are independent of the predicted overall decision, the authors define an explanation factor that helps build stronger correlations between the explanations and the predictions. More specifically, a classifier C is trained to predict the category from the explanations and not the original input text. C is then used to provide more robust guidance for the text encoder to leverage the generation process by generating a more informative representation vector v_e . Experiments show that the explanation factor enhances the performance of the base predictor model.

Reasoning over Knowledge Graph [97] presents a promising way to explain NLP systems in a structured way in the form of $\langle \text{source_entity}, \text{relation}, \text{target_entity} \rangle$ [117]. Graph reasoning has been addressed as a variational inference problem [17], random-walk search [36], and a Markov Decision Processes [24, 120]. Recently, knowledge graph reasoning has attracted researchers in the NLP and ExAI communities. Readers are referred to Reference [18] where its basic concept, definitions, and methods are surveyed.

8.3 Discussion

To the best of our knowledge, *inherently* interpretable models for explaining individual decisions of NLP models are restricted to the work of References [55, 61]. This can be explained by the fact that current language models require large computational costs in terms of time and computational resources. Modifying the architecture to an explainable one and retraining the current high-performance models will be expensive.

On the evaluation level, since *model-agnostic* approaches were not specific to textual data, their evaluation was general and not specific to NLP cases. For instance, LIME's evaluation for textual data was done qualitatively by getting insights on a model trained to differentiate "Christianity" from "Atheism" on a subset of the 20 newsgroup dataset [88]. However, SHAP [63] is not explicitly tested on NLP tasks, and References [89]'s anchors were tested on tabular datasets. Although a common evaluation framework is not followed with the *model-specific* approaches, textual data are used to evaluate the described explainability methods. This evaluation is inspired by sensitivity analysis where the sentence representation is monitored to evaluate how much deviation is observed when the words outputted by the explainability method are omitted [5, 51]. Regarding the *inherently* interpretable models, one can clearly see that both methods [55, 61] rely on the encoder-decoder design to generate explanations. Reference [55] evaluates their approach on multi-aspect sentiment analysis and compare their explanations to manually annotated test cases. Although Reference [61] has a similar motivation, it does not compare to Reference [55]. The authors report instead a BLEU score for their generated explanations. They also report the performance of a classifier on the fine-grained explanation instead of the initial input and employ a human evaluation framework.

9 DISCUSSION AND FUTURE DIRECTIONS

To better understand the interest in ExAI in NLP, we consider the titles of the papers that we reference in this work, and we visualize word frequencies in Figure 5. One can see that *deep networks*,

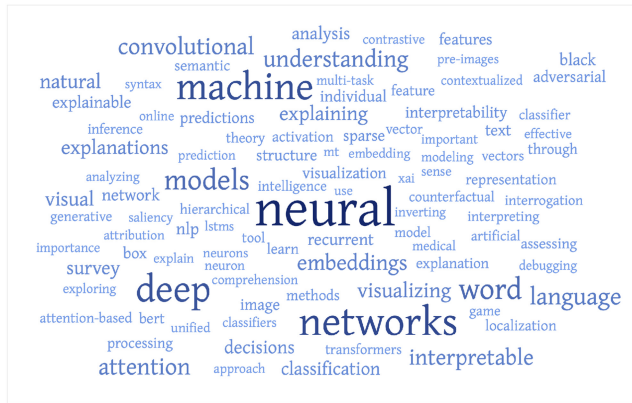


Fig. 5. Most frequent terms used in the titles of the publications referenced in this work.

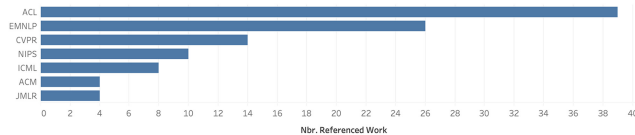


Fig. 6. Top 7 conferences and journals referenced in this work.

representation or embedding methods, and attention models are very frequent in our referenced work. The importance of visualizations or visual clarification techniques in explaining deep models can be also inferred from the frequencies. The figure also hints at the fact that ExAI in NLP in the referenced work is mostly focused on *understanding* the inner workings of the underlying models rather than understanding a particular output of *classification*. Additionally, to reflect the interest in ExAI in general within the machine learning community, we show the top 7 conferences and journals referenced in this work in Figure 6. The lack of journals that study or survey ExAI methods in general, and in NLP in particular, is reflected in the figure. Researchers are publishing their ExAI methods and discoveries in general AI conferences or conferences that focus on linguistics and natural language.

Furthermore, Figure 7 shows how the research interest in some ExAI categories has been progressing since 2012. For instance, the interpretability of word embeddings has attracted researchers since 2015, when the concern about bias in the machine learning model has emerged after Google Photos application tagged a black woman as Gorilla [38]. The introduction of transformers in 2017 has also encouraged researchers to study the magic behind their state-of-the-art performance on different NLP tasks. Moreover, the unprecedented breakthroughs in NMT in 2016 [119] encouraged research work on ExAI applied to NMT models afterward.

Although recent years have shown significant growth in ExAI methods applied to NLP models, these methods are not fulfilling their potential yet. For instance, the evaluation of ExAI methods is lacking a unique testing framework where metrics and datasets are well designed. In the assessment of ExAI’s current methods, different NLP tasks such as POS tagging, word intrusion, and correlation experiments are explored. However, these tasks are paper specific, and they are not used to compare different methods. This brings to the forefront the need for a common evaluation framework where human-labeled datasets are well defined and NLP tasks are described within the syntactic or semantic aspect that they aim to explain. Additionally, an area that is yet to be

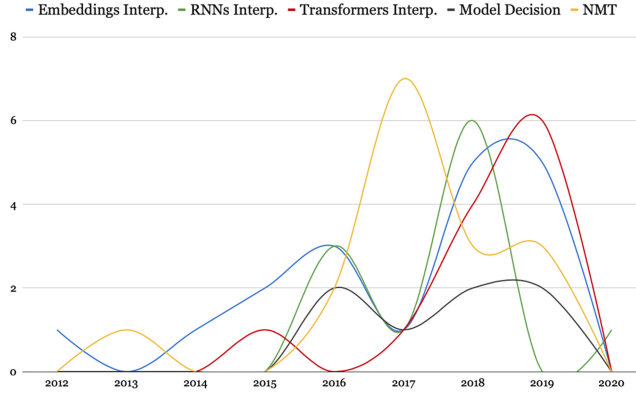


Fig. 7. Timeline for ExAI in NLP.

explored by researchers is the quantitative assessment of the interpretability of the embedding spaces and the effectiveness of each dimension in semantically and syntactically encoding a particular concept.

Moreover, the majority of the work studied in this survey post-process NLP models to explain their decisions and inner workings. *Inherently* interpretable NLP models are under-examined due to the fact that retraining a model after modifying its architecture is very expensive and might not achieve the same performance. Exploring explainable designs for NLP models can thus be one of the main subjects for future work. Another direction for future work is the level at which explanations for language models are provided. So far, the explanations are provided as either the contribution of individual words to the decision or the layer/neuron at which syntax or semantics are encoded. However, text analysis is a multi-step process: After extracting information from textual unstructured data, analysis is applied to the extracted information to reach knowledge. Wisdom and logic come at the highest level of semantics. Thus, explanations provided on the individual input words will be ignoring the hierarchy of the text understanding process, which will affect their efficacy.

10 CONCLUSION

This work presents the first comprehensive survey on explainability methods in the NLP field that combines ExAI methods on the input, processing, and output levels. According to the assumptions made on explained models, we make the distinction between *model-agnostic* and *model-specific* methods. We further distinguish between explanation in a *post hoc* manner and explanation that results in *inherently* interpretable models.

We present different attempts to interpret word embedding models that have recently served as inputs to almost every NLP network. Some of those methods rely on altering the embedding space by imposing a sparsity constraint or applying a rotation transformation, while others integrate external knowledge bases and ontologies or rely on bidirectional language models to derive contextualized embeddings. Moreover, we survey existing work on the interpretation of hidden representations of NLP models, general RNNs, and transformers in terms of human-understandable concepts. We discuss the debate over the interpretability of attention weights, and we derive the following conclusion: Attention weights are relatively more *inherently* interpretable than traditional models' parameters, but further analysis is required for complete transparency in attention-based models. Additionally, we present the research work that explains a particular model decision

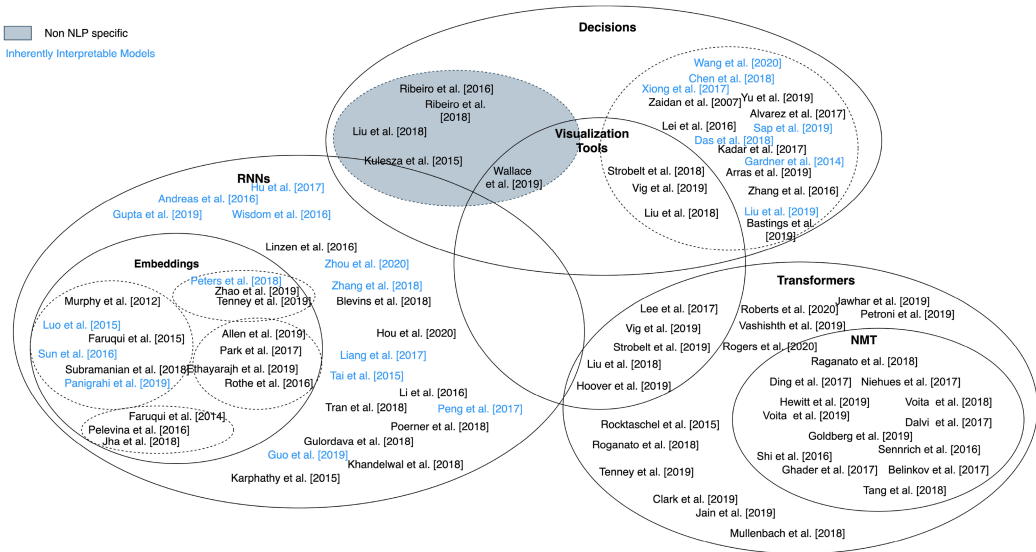


Fig. 8. References for Sections 5, 6, and 7 visualized over similarity of scopes.

inspired by general ExAI methods or designed specifically for NLP models. We also discuss different visualization platforms that present user-friendly explainability schemes based on perturbations and attention weights.

Figure 8 summarizes the work done on the interpretability of word embeddings, inner workings of RNNs and transformers, the model’s decision, and the different visualization methods while highlighting the interconnections between the different methods.

To date, there is no common evaluation ground for explainability methods on NLP models. In this work, we shed the light on different empirical setups, datasets, and metrics that can be designed to assess the performance of ExAI in NLP. These setups are aggregated according to *what* the corresponding ExAI method is addressing: embeddings, inner workings, or model’s decisions. We further examine these evaluation methods to discriminate between evaluations that explicitly assess the interpretability and those that are of qualitative nature or need further analysis to extract insights useful from an explainability perspective. Finally, we discuss the limitations of existing ExAI methods while highlighting research areas that can possibly be the focus of researchers in their future exploration.

REFERENCES

- [1] Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the International Conference on Machine Learning*. 223–231.
- [2] David Alvarez-Melis and Tommi S. Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP’17)*.
- [3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1545–1554.
- [4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.
- [5] Leila Arras, José Arjona-Medina, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter, and Wojciech Samek. 2019. Explaining and interpreting LSTMs. In *Explainable ai: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 211–238.

- [6] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One* 10, 7 (2015).
- [7] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.* 11 (2010), 1803–1831.
- [8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473. Retrieved from <https://arxiv.org/abs/1409.0473>.
- [9] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On identifiability in transformers. In *International Conference on Learning Representations*.
- [10] Joost Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL Anthology, 2963–2977.
- [11] Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 14–19.
- [12] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*. 4349–4357.
- [13] Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. [n.d.]. On Identifiability in Transformers.
- [14] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1448–1458.
- [15] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the International Conference on Machine Learning*. PMLR, 883–892.
- [16] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. arXiv:1808.02610. Retrieved from <https://arxiv.org/abs/1808.02610>.
- [17] Wenhui Chen, Wenhui Xiong, Xifeng Yan, and William Yang Wang. 2018. Variational knowledge graph reasoning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1823–1832.
- [18] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* 141 (2020), 112948.
- [19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT’s attention. In *Proceedings of the ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 276–286.
- [20] Charles B. Crawford and George A. Ferguson. 1970. A general rotation criterion and its use in orthogonal rotation. *Psychometrika* 35, 3 (1970), 321–332.
- [21] Piotr Dabkowski and Yarin Gal. 2017. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*. 6967–6976.
- [22] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. arXiv:2010.00711. Retrieved from <https://arxiv.org/abs/2010.00711>.
- [23] Arun Das and Paul Rad. 2020. Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv:2006.11371. Retrieved from <https://arxiv.org/abs/2006.11371>.
- [24] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.
- [25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [26] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4443–4458.
- [27] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1150–1159.
- [28] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. arXiv:1702.08608. Retrieved from <https://arxiv.org/abs/1702.08608>.

- [29] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable artificial intelligence: A survey. In *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MICRO'18)*. IEEE, 0210–0215.
- [30] Alexey Dosovitskiy and Thomas Brox. 2015. Inverting convolutional networks with convolutional networks. arXiv:1506.02753. Retrieved from <https://arxiv.org/abs/1506.02753>.
- [31] Philipp Dufter and Hinrich Schütze. 2019. Analytical methods for interpretable ultradense word embeddings. arXiv:1904.08654. Retrieved from <https://arxiv.org/abs/1904.08654>.
- [32] Kawin Ethayarajh. 2019. Rotate king to get queen: Word relationships as orthogonal transformations in embedding space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 3494–3499.
- [33] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1606–1615.
- [34] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 1: Long Papers)*. 1491–1500.
- [35] Bent Fuglede and Flemming Topsøe. 2004. Jensen-shannon divergence and hilbert space embedding. In *Proceedings of the International Symposium on Information Theory (ISIT'04)*. IEEE, 31.
- [36] Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 397–406.
- [37] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings of the IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA'18)*. IEEE, 80–89.
- [38] Loren Grush. 2015. Google Engineer Apologizes After Photos App Tags Two Black People as Gorillas. Retrieved September 28, 2020 from <https://www.theverge.com/2015/7/1/8880363/google-apologizes-photos-app-tags-two-black-people-gorillas>.
- [39] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Comput. Surv.* 51, 5 (2018), 1–42.
- [40] Kristina Gulordava, Piotr Bojanowski, Édouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1195–1205.
- [41] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 241–251.
- [42] Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. arXiv:1912.04971. Retrieved from <https://arxiv.org/abs/1912.04971>.
- [43] John R. Hershey, Jonathan Le Roux, and Felix Weninger. 2014. Deep unfolding: Model-based inspiration of novel deep architectures. arXiv:1409.2574. Retrieved from <https://arxiv.org/abs/1409.2574>.
- [44] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. arXiv:1910.05276. Retrieved from <https://arxiv.org/abs/1910.05276>.
- [45] Bo-Jian Hou and Zhi-Hua Zhou. 2020. Learning with interpretable structure from gated RNN. *IEEE Trans. Neural Netw. Learn. Syst.* (2020).
- [46] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. 804–813.
- [47] Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 3543–3556.
- [48] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*.
- [49] Kishlay Jha, Yaqing Wang, Guangxu Xun, and Aidong Zhang. 2018. Interpretable word embeddings for medical domain. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'18)*. IEEE, 1061–1066.
- [50] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Trans. Assoc. Comput. Ling.* 8 (2020), 423–438.
- [51] Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Comput. Ling.* 43, 4 (2017), 761–780.

- [52] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. arXiv:1506.02078. Retrieved from <https://arxiv.org/abs/1503.02078>.
- [53] Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 284–294.
- [54] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 121–126.
- [55] Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*.
- [56] Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of NLP models: A survey. arXiv:2104.15135. Retrieved from <https://arxiv.org/abs/2104.15135>.
- [57] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'16)*. 681–691.
- [58] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. arXiv:1612.08220. Retrieved from <https://arxiv.org/abs/1312.08220>.
- [59] Xiaodan Liang, Liang Lin, Xiaohui Shen, Jiashi Feng, Shuicheng Yan, and Eric P. Xing. 2017. Interpretable structure-evolving LSTM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1010–1019.
- [60] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Ling.* 4 (2016), 521–535.
- [61] Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5570–5581.
- [62] Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. *Visual Interrogation of Attention-based Models for Natural Language Inference and Machine Comprehension*. Technical Report. Lawrence Livermore National Laboratory, Livermore, CA.
- [63] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 4765–4774.
- [64] Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1687–1692.
- [65] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (November 2008), 2579–2605.
- [66] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5188–5196.
- [67] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781. Retrieved from <https://arxiv.org/abs/1301.3781>.
- [68] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267 (2019), 1–38.
- [69] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasani Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4206–4216.
- [70] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digit. Sign. Process.* 73 (2018), 1–15.
- [71] Raymond Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'96)*.
- [72] James Mullenbach, Sarah Wiegrefe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1101–1111.
- [73] Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the International Conference on Computational Linguistics (COLING'12)*. 1933–1950.
- [74] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2019. Understanding neural networks via feature visualization: A survey. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 55–76.
- [75] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2Sense: Sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5692–5705.

- [76] Sungjoon Park, JinYeong Bak, and Alice Oh. 2017. Rotated word vector representations and their interpretability. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 401–411.
- [77] Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. 174–183.
- [78] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. Comput. Ling.* 5 (2017), 101–115.
- [79] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.
- [80] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1499–1509.
- [81] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'18)*. 2227–2237.
- [82] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 2463–2473.
- [83] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4782–4793.
- [84] Erika Puiutta and Eric Veith. 2020. Explainable reinforcement learning: A survey. arXiv:2005.06247. Retrieved from <https://arxiv.org/abs/2005.06247>.
- [85] Zhuwei Qin, Fuxun Yu, Chenchen Liu, and Xiang Chen. 2018. How convolutional neural networks see the world—a survey of convolutional neural network visualization methods. *Math. Found. Comput.* 1, 2 (2018), 149.
- [86] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
- [87] Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 287–297.
- [88] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.
- [89] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- [90] Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 5418–5426.
- [91] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. arXiv:1509.06664. Retrieved from <https://arxiv.org/abs/1509.06664>.
- [92] Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumshisky. 2018. What’s in your embedding, and how it predicts task performance. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2690–2703.
- [93] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Trans. Assoc. Comput. Ling.* 8 (2020), 842–866.
- [94] Alexis Ross, Ana Marasović, and Matthew E. Peters. 2020. Explaining nlp models via minimal contrastive editing (mice). arXiv:2012.13985. Retrieved from <https://arxiv.org/abs/2012.13985>.
- [95] Sascha Rothe and Hinrich Schütze. 2016. Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 512–517.
- [96] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* 1, 5 (2019), 206–215.
- [97] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3027–3035.
- [98] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.

- [99] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 3145–3153.
- [100] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.
- [101] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. 2018. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2018), 353–363.
- [102] Emma Strubell, Anya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. arXiv:1906.02243. Retrieved from <https://arxiv.org/abs/1906.02243>.
- [103] Erik Strumbelj and Igor Kononenko. 2010. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.* 11 (2010), 1–18.
- [104] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- [105] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Sparse word embeddings using l1 regularized online learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 2915–2921.
- [106] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1556–1566.
- [107] Oguzhan Tas and Farzad Kiyani. 2007. A survey automatic text summarization. *PressAcademia Procedia* 5, 1 (2007), 205–213.
- [108] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4593–4601.
- [109] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, et al. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. arXiv:1905.06316. Retrieved from <https://arxiv.org/abs/1905.06313>.
- [110] Erico Tjoa and Cuntai Guan. 2019. A survey on explainable artificial intelligence (XAI): Towards medical XAI. CoRR abs/1907.07374 (2019).
- [111] Ke M. Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 4731–4736.
- [112] Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. arXiv:1909.11218. Retrieved from <https://arxiv.org/abs/1909.11218>.
- [113] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. 193–199.
- [114] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
- [115] Jesse Vig. 2019. Visualizing attention in transformer-based language representation models. arXiv:1904.02679. Retrieved from <https://arxiv.org/abs/1904.02679>.
- [116] Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 63–76.
- [117] Cunxiang Wang, Jinhang Wu, Luxin Liu, and Yue Zhang. 2020. Commonsense knowledge graph reasoning by selection or generation? Why? arXiv:2008.05925. Retrieved from <https://arxiv.org/abs/2008.05925>.
- [118] Scott Wisdom, Thomas Powers, James Pitton, and Les Atlas. 2016. Interpretable recurrent neural networks using sequential sparse recovery. arXiv:1611.07252. Retrieved from <https://arxiv.org/abs/1611.07252>.
- [119] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144. Retrieved from <https://arxiv.org/abs/1609.08144>.
- [120] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 564–573.
- [121] Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *Proceedings of the European Conference on Computer Vision*. Springer, 451–466.
- [122] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 21–29.

- [123] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4094–4103.
- [124] Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using “annotator rationales” to improve machine learning for text categorization. In *Proceedings of the Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*. 260–267.
- [125] Ye Zhang, Iain Marshall, and Byron C. Wallace. 2016. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Vol. 2016. NIH Public Access, 795.
- [126] Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1554–1564.
- [127] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. arXiv:1904.03310 [cs.CL]. Retrieved from <https://arxiv.org/abs/1904.03310>.
- [128] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 4847–4853.
- [129] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

Received 2 April 2021; revised 4 October 2021; accepted 30 March 2022