Challenging SRL models report

Anonymous ACL submission

1 Introduction

1.1 Motivation and approach

In common NLP (Natural Language Processing) tasks, the standard evaluation method is using train-validation-test splits to estimate the accuracy of the model. However, the data used for accuracy tests often contain the same biases as in the training data, and most of the time, incomprehensive (Ribeiro, 2020). Therefore, overall model performances usually cannot be precisely estimated and are often overestimated. In addition, a single aggregate evaluation statistic has limitations.

Software engineering has proposed a variety of paradigms and tools for testing complex software systems, in which 'behavioral testing' (also known as black-box testing) is one of them. It is concerned with testing different capabilities of a system by validating the input-output behavior, without any prior knowledge of the internal structure (Beizer, 1995).

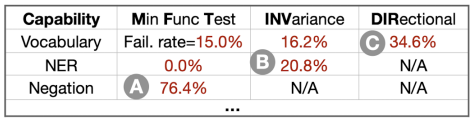
1.2 Checklist

Ribeiro (2020) applies the insights from software engineering to NLP models and proposes CHECKLIST, which is a new evaluation methodology and accompanying tool for comprehensive behavioral testing of NLP models. Three NLP tasks, sentiment analysis (Sentiment), duplicate question detection (QQP), and machine comprehension (MC) are used to demonstrate the usefulness and generality of CHECKLIST. For Sentiment, a possible approach is to test if the model is able to recognize words that carry positive, neutral, or negative sentiment by verifying the model behavior on examples like 'This was a nice meal.' For QQP, the model is expected to understand when modifiers differentiate questions. For MC, the model should be able to connect comparatives and superlatives, e.g. (Context: 'Lisa is smarter than Maggie.', Q: 'Who is the smartest kid?', A: 'Lisa').

Three test types Furthermore, Ribeiro (2020) provides three different test types (when possible) for users to evaluate each capability: 1) Minimum Functionality test (MFT) - inspired by unit tests in software engineering, is a collection of simple examples (and labels) to check a behavior within a capability. 2) Invariance test (INV) - is when we apply label-preserving perturbations to inputs and expect the model prediction to remain the same. 3) Directional Expectation test (DIR) - is when we expect the label to change in a certain way.

Checklist matrix In a checklist matrix, potential tests are structured conceptually (see figure 1), with capabilities as rows and test types as columns. Not every capability has all three test types (MFT, INV, DIR).

Figure 1: Checklist matrix example (image taken from Ribeiro (2020)

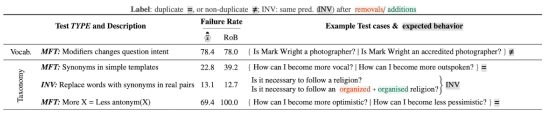


Ribeiro (2020) provided selections of three task specific checklists, which are tests for sentiment analysis, Quora Question Pair and Machine Comprehension.

In tests for Quora Question Pair, the apabilities of Vocab, Taxonomy, Robust, NER, Temporal, Negation, Coreference, SRL and Logic were tested by utilising all three test types. For each capability, test type and description, the failure rate of each model, example test cases and expected behavior are presented (see figure 2). One of their findings from the Checklist tests is that both models lack

crucial capabilities for the task, such as ignoring important modifiers on Vocab test, and lacking basic Taxonomy understanding.

Figure 2: A selection of tests for Quora Question Pair (image taken from Ribeiro (2020)



Generating test instances To create test instances, users can start from scratch by creating a small number of high-quality test cases for specific, underestimated and confounded phenomena in the original dataset. However, the downside is that it requires creativity and effort yet still has problems such as low coverage, high cost and time-consuming.

The other way of generating test instances is to perturb an existing dataset. Users can create their own perturbation functions, which aim to generate many test cases at once. A variety of abstractions, which scale up test creation from scratch and make perturbations easier to craft, were provided in Ribeiro (2020).

2 Background

2.1 Semantic Role labeling

Semantic parsing of sentences is believed to be an important task on the road to natural language understanding, and has immediate applications in tasks such as information extraction and question answering. Semantic role labelling (SRL) is a shallow semantic parsing task, in which for each predicate in a sentence, the goal is to identify all constituents that fill a semantic role, and to determine their roles (Agent, Patient, Instruments, etc.) and their adjuncts (Locative, Temporal, Manner, etc.) (Punyakanok, 2008).

Jurafsky and Martin (2019) points out that SRL can be treated in the general framework of a classification task: for a given verb, and given each constituent in a parse, the task is to select from a pre-defined set the constituent's semantic role label with respect to the verb. Current approaches to SRL are based on supervised machine learning, often using the FrameNet and PropBank resources to specify what counts as a predicate, define the set of roles used in the task, and provide training and test sets. The algorithms generally start by parsing a sentence and then automatically tag each parse tree node with a semantic role.

3 Core capabilities of SRL

3.1 Identify high position arguments

In Propbank SRL, arg2 - 5 are verb specific roles, are usually variable and overloaded, and usually poor performing labels [1]. Thus, we would expect a SRL model to be able to identify multiple argument structures.

> *I'm sorry to say Elena's story has been revealed to be a fake.*

In the above example, 'fake' is labeled as ARG3[2] of the predicate 'revealed'.

3.2 Causative inchoative alternation

The causative-inchoative lateration is a lexical alternation. It characterizes pairs of verbs that stand in approximately the following semantic relation to each other: the intransitive member of the pair, a.k.a an inchoative verb, denotes a change of state, and the transitive member of the pair, a.k.a. a causative-inchoative verb, denotes a bringing about of this change of state. (Piñón, 2001). The sentences below illustrate the causative-inchoative alternation with typical pairs of alternating verbs.

> *1) a. Rebecca broke the pencil.*
> *b. The pencil broke.*
> *c. The pencil was broke by Rebecca.*
> *2) a. Maria opened the door.*
> *b. The door opened.*
> *3) a. Thomas dried the clothes.*
> *b. The clothes dried.*

We expect SRL systems to be able to assign the correct labels to ARG0 and ARG1. For example, in the first example, 'pencil' should be the proto-patient ARG1, and 'Rebecca' should be the proto-agent ARG0.

3.3 Subordinate clause

A subordinate clause is a clause that is embedded within a complex sentence. A verb within a subordinating clause usually has the subordinating conjunction as its dependency head and it tends to be long-range dependency, which is harder to predict.

[1] source: Advanced NLP lecture2 slides

[2] ARG3-PRD: attribute of arg1; source: https://propbank.github.io/v3.4.0/frames/reveal.html

The difficulty comes from the complexity of the sentence and the amount of parts.

For example, *that* in 'I thought that you might like some milk' is a subordinating conjunction that links the main clause *I thought* with the subordinate clause *you might like some milk*.

3.4 Long-span dependencies

In a sentence of long-span dependencies, the distance span of the predicate and its argument can be relatively large, which could potentially bring out difficulties for SRL models to parse correctly. For example, in the sentence 'Two hundred ***members*** of the Batawi clan of the Dulaim demonstrated in Baghdad on Friday, **protesting** the killing of their clan elder, Shaikh Kadhim Sarhid and 4 of his sons, by gunmen wearing Iraqi army uniforms', the head word 'members' should be labeled as ARG0 of the predicate 'protesting'.

3.5 Voice

Palmer et al. (2010): Direct objects of active verbs often correspond in semantic role to subjects of passive verbs. Therefore, the distinction between active and passive verbs plays a significant role in the connection between semantic role and grammatical function. In English, passive expressions are prevalent. Roughly 5% of the FrameNet examples were identified as passive uses. Roland (2001) reports that 6.7% of verbs are passive in the Penn Treebank Wall Street Journal corpus, and 7.8% in the Brown corpus. Consider the below two sentences:

He conducted the experiment.
The experiment was conducted by him.

In this example, the structure of the first active sentence is 'subject + verb + object', while the second passive one is 'object + verb (passive form) + by (proposition) + subject'. We would expect SRL models to be able to learn differences of voice, since it is a prevalent phenomenon, and always assign the right subject and object as ARG0 and ARG1.

3.6 NER

Named entities usually refer to words of classes person, organization, location, percent, money, time and date (Palmer et al., 2010). Named entity recognition itself is an important task for NLP. It helps handle the data sparsity caused by the unlimited sets of proper names in particular for people, organizations and locations (Palmer et al., 2010). We are expecting SRL models to have the capability in understanding named entities appropriately by assigning the correct label to the exact span or head of named entities. For example, in the sentence 'President Bush nominated three candidates.', 'President Bush' should be recognized as ARG0 of the predicate 'nominated'.

3.7 POS

A predicate can be a verb or a noun. In propbank, many frames have several aliases containing nouns and verbs. For example, collect.01 has three aliases: collecting (n.), collection (n.) and collect (v.). In the sentence 'He has a large toy collection', we would expect SRL models to correctly assign ARG1 to 'toy' of the noun predicate 'collection'.

3.8 PP attachment

In English, the sentence 'I saw the kid with the cat' has two potential parsing. The first one is that the propositional phrase attached high, modifying the verb 'saw', and the second one is that it attached low, to the object 'the kid'. There is also a difference in the sense of the preposition (instrumental vs. comitative) but often the ambiguity, which is called PP attachment ambiguity, is structural. In English NP attachment is slightly more common in English, thus some probability-based parsing systems would have the tendency preferring NP attachment [ARG0: I] [V: saw] [ARG1: the kid with a cat] [saw [the kid with the cat]'. Another example:

I killed him with a knife.

In this case, we would expect a parsing like this: [ARG0: I] [V: killed] [ARG1: the kid] [ARG2: with a knife], which would make more sense semantically linking the predicate 'killed' and the instrument 'knife'. For the capability of identifying PP attachment ambiguity, we would expect SRL models could disambiguate this ambiguity by learning context semantics as humans do.

3.9 Robustness

We will focus on typo and negation for the robustness test. Negation phenomena is a core capability of sentiment analysis, but in SRL, negating a verb should not induce changes in argument labels. For example, if one instance has random and irrelevant

information such as an URL, the model should be able to produce the same result for each argument regardless of whether the URL presenting or not. In addition, for sentence with or without negation, such as 'I do like him' and 'I don't like him', we expect the model to produce the same label for 'him' of the predicate 'like'.

4 Challenging Dataset

4.1 What is challenge dataset

As we mentioned in the first section, the data used for accuracy tests often contain biases. Belinkov (2019) point out that most benchmark datasets in NLP are drawn from text corpora, reflecting the natural frequency distribution of linguistic phenomena. However, such datasets may fail to capture a variety of phenomena. Challenge sets, also known as test suites, are alternative evaluation frameworks that have been used in NLP for a long time. Some key properties of challenge sets are systematicity, control over data, inclusion of negative data, and exhaustivity, which contrast with benchmark text corpora, 'whose main advantage is to reflect naturally occurring data'. Usually, the below criteria are used to categorize the challenge datasets: 1) the task they seek to evaluate, 2) the linguistic phenomena they aim to study, 3) the language(s) they target, 4) their size, 5) their method of construction, 6) how the performance is evaluated.

4.2 Challenge dataset for SRL

To evaluate various capabilities of SRL models, we implemented some of the proposed tests we mentioned in section 3 by creating a challenge dataset.

4.2.1 Resources utlized

Here we introduce the resources we utilized to create our challenge dataset:

Universal Proposition Databanks Most of the test instances of our challenging dataset were created manually assisted by the training dataset from the Universal Proposition Databanks [3] (version 1.0, in CoNLL-U Plus format, hereinafter referred to as 'reference dataset'). We first converted this training dataset to a Json file, each instance in which contain a word sequence, label sequence, and predicate information (position, sense, lemma). Then we loaded the json file into a Pandas DataFrame and used Boolean Indexing with relevant conditions corresponding to each capability to get the test instances. In addition, part of the instances were retrieved from relevant literature.

PropBank To generate an accurate label for the head word argument, we referred to the PropBank frames extensively [4].

4.2.2 capability test design

Test 1: Identify high position arguments usually SRL models perform well in predicting ARG0 and ARG1 than ARG2-5 since the former (proto-agent and proto-patient) are more directly linked to the predicate. To test this capability, we will implement several MFT tests on sentences containing ARG3 or ARG4. For example, for verbs buy/purchase/sell, their arg3 are 'price paid'; for verb 'smear', its arg3 is instrument. In addition, in our challenging dataset, we have this sentence: stretching 750 kilometers from end(1) to end(2) , they reach from near the coast of Myanmar almost to Sumatra in Indonesia. Here we expect the model to predict 'end'(2) as ARG4 (GOL: end point).

To find the test instances containing ARG3 and ARG4, we referred to examples in the reference dataset.

Test 2: Causative inchoative alternation

to find examples causative inchoative alternation, we consulted Piñón (2001), as well as utlilize online resources like Wikipedia and ChatGPT by OpenAI. At last we designed 8 INV test cases for this capability including verbs such as 'break', 'dry', 'open', 'freeze', etc, comparing the ARG1 of the inchoative verb and the causative-inchotive verb.

Test 3: Subordinate clause to create subordinate clause examples, we looked for the instances that have subordinating conjunctions like 'that' (cleft clause), 'whehter' and 'which' in the training dataset of Universal Proposition Databanks, and manually selected examples for our challenge dataset.

Test 4: Long-span dependencies to find sentences with long-span dependencies, we filtered

[3] https://github.com/UniversalPropositions/UP-1.0/tree/master/UP_English-EWT

[4] https://propbank.github.io/v3.4.0/frames/

out instances that contained at least 15 tokens and did a manual selection.

Test 5: Voice we designed 5 INV sentence pairs to test models' capability handling active/passive voice. In particular, we designed test cases like this:

> *The passenger killed the killer.*
> *The killer was killed by the passenger.*

Usually, when 'killer' co-occurs with the predicate 'kill', its role is ARG0. However, in this case, to test if our model's capability of detecting the difference between voices, we designed 'killer' as the ARG1. If a model is able to assign the correct label, then it will demonstrate its capability in learning the syntactic structure not just by a shortcut that always connects 'killer' with ARG0.

Test 6: NER we utilized the reference dataset, selected and designed 5 INV test cases that contain named-entities of organizations, location, time and person.

Test 7: POS as we mentioned in the section 2, both verb and noun can act as predicates. To craft test cases, we scanned through frames in PropBank, in which we looked for aliases that has noun form. At last we designed 5 MFT test cases that include noun predicates 'nomination', 'cooperation', 'discovery', 'collection' and 'advance' to test the POS capability.

Test 8: Robustness We designed 5 INV test cases to test the robustness of SRL models including sentence pairs with negation, typos and irrelevant random information.

4.2.3 Evaluation criteria

In Lecture 2 slide 52 presented two evaluation approaches for SRL models: full argument span vs. head word argument, the latter of which always has higher precision, recall and f1-score. To evaluate SRL models on our challenge dataset, we choose to implement head word argument evaluation mostly for the reason of simplicity and efficiency. The simplicity here refers to the convenience of creating the challenge dataset, and the conciseness of the dataset structure. We believe in this initial challenging SRL models experiment, using the head word for evaluation is sufficient to investigate models' capabilities in handling linguistics phenomena.

5 Models

In this challenging SRL models experiment, we evaluated two SRL models provided by AllenNLP. The first model is a BERT-based model with some modifications. AllenNLP trained and evaluated this model on the Ontonotes 5.0 dataset.[5] Below is its introduction from AllenNLP's website[6]:

> *An implementation of a BERT-based model (Shi et al, 2019) with some modifications (no additional parameters apart from a linear classification layer), which is currently the state of the art single model for English PropBank SRL (Newswire sentences). It achieves 86.49 test F1 on the Ontonotes 5.0 dataset.*

The second model is a LSTM model that is a re-implementation of a deep Bi-LSTM sequence prediction model (Stanovsky et al., 2018). It is a supervised model for Open Information Extraction (Open IE), the central approach of which is a novel formulation of Open IE as a sequence tagging problem addressing challenges such as encoding multiple extractions for a predicate.

We would expect that the first BERT-based model would outperform LSTM model since it is built upon a self-attention mechanism, allowing modeling of dependencies without regard to their distance in the input or output sequences.

6 Results

Table 1 and Table 2 are two checklist metrics for the two models we evaluated on. Overall, we conclude that the BERT-based SRL model performed better on our challenge dataset than Bi-LSTM SRL model, since its failure rates of each capability are either equal to or lower than the latter model, which is in line with our expectation that the BERT-based model would outperform Bi-LSTM model. In 36 cases that the BERT-based model passed, Bi-LSTM failed 3. In 34 cases that Bi-LSTM passed, the BERT model failed 1. These indicate that these two models may have an overlap in learning capabilities.

[5] https://catalog.ldc.upenn.edu/LDC2013T19
[6] https://demo.allennlp.org/semantic-role-labeling

Capability	Min Func Test	INVariance
Identify high position argument	50%	N/A
Long-span dependencies	20%	N/A
Subordinate clause	0	N/A
Causative Inchoative Alternation	N/A	0
POS	100%	N/A
NER	0	N/A
Robustness	N/A	0
Voice	N/A	0

Table 1: Checklist matrix: Fail-rate BERT-based SRL model

Capability	Min Func Test	INVariance
Identify high position argument	62.5%	N/A
Long-span dependencies	20%	N/A
Subordinate clause	0	N/A
Causative Inchoative Alternation	N/A	12.5%
POS	100%	N/A
NER	0	N/A
Robustness	N/A	0
Voice	N/A	0

Table 2: Checklist matrix: Fail-rate Bi-LSTM SRL model

6.1 Identify high position arguments

Test case (MFT) (**predicate** , ***token for evaluation***)	Gold
1) Thanks for the **comment** on the ***hearing***.	ARG3
2) **Stretching** 750 kilometers from end to ***end***, they reach from near the coast of Myanmar almost to Sumatra in Indonesia.	ARG4
3) There are a lot of genetics at play between a Sussex and Silkie, so the offspring would **vary** in ***appearance***.	ARG5

For BERT-based model, among 9 test cases, 4 pass and 1 case is invalid (see sentence 1) because the model was not able to recognize the noun predicate 'comment'. One problem we find of identifying high position arguments is that the model tend to label high position arguments as adjunct arguments. In example 2), the model predicted 'end' as ARGM-TMP in the constituent 'from end to end', and in example 3), 'appearance' was labeled as 'ARGM-MNR', while the gold label, according to the roleset of 'vary', should be 'ARG5-PRD': medium'[7].

For the Bi-LSTM model, among 9 test cases, 3 pass and 1 case, as the BERT-based model, is invalid because the model was not able to recognize the noun predicate comment. The model also have share the same tendency of labeling high position arguments as adjunct arguments, and made one more misprediction than the BERT-based model.

We interpret these results from two aspects. Firstly, high position arguments are in nature hard to identify because they are usually verb-specific roles, which means they have more variations depending on the specific predicate, and some of them share the same syntactic structure with adjunct arguments, which also leads to misprediction. Secondly, we argue that there's an existing ambiguity of semantic role annotation standards. In example 3), the gold label 'ARG5-PRD: medium' and the predicted label 'ARGM-MNR' seem all reasonable to label '[V: vary] [in appearance]'.

6.2 Subordinate Clause

Both BERT and Bi-LSTM models have passed all 5 MFT tests, assigning the labels we expected to the correct constituents that contain head words we evaluated on. From these tests, we conclude that both models have basic capability of successfully identify the relationship between the predicate and the argument located on either side of subordinating conjunctions. We interpret the success from two aspects. First, there are limited amount of subordinating conjunctions existing in English, which results in less variation. Second, the test cases we designed are fairly short sentences, and we were only testing on low position arguments (ARG0 and ARG1), which also contributed to the successful identification.

[7] https://propbank.github.io/v3.4.0/frames/vary.html

6.3 Long-span dependencies

Test case (MFT) (**predicate** , ***token for evaluation***)	Gold ARG1
In the last decade, there has been a real and significant ***increase*** in childhood and, to a certain extent, adult carcinoma of the thyroid in contaminated regions of the former Soviet Union (Wi940) which should be **attributed** to the Chernobyl accident until proven otherwise	

Both models passed the same 4 and failed 1 MFT long-span dependencies test case. In the failing case (presented above), the span between the predicate and the argument constituent is more than 20 tokens.

6.4 Causative Inchoative Alternation

The BERT-based SRL model passed all 7 INV causative inchoative alternation test cases, showing the ability to identify alternating verbs and assign the correct labels. Meanwhile, the Bi-LSTM model failed in one test case 'The water froze quickly' in which it assigned ARG0 to the noun phrase 'the water'.

6.5 POS

Test case (MFT) (**predicate** , ***token for evaluation***)	Gold ARG0
Thanks in advance for ***your*** **cooperation**.	
The utilities repeatedly called on FERC to do a "real" investigation, with hearings, testimony, ***data*** **discovery** — the works.	ARG1

Both models failed all 5 MFT POS tests. We interpret this failure from two aspects: firstly, we speculate that the annotation datasets used for training SRL models might lack instances in which nouns are predicates. Secondly, noun predicates have different syntactic behavior with most verb predicates.

6.6 NER

Both models passed all 5 MFT NER tests, correctly identifying tokens with expected labels. In addition, both models assigned all NER constituents with the correct span.

6.7 VOICE

Both models passed all INV voice test cases, indicating that they have the basic capability to identify passive voice in English.

6.8 Robustness

Both models passed all INV test cases with random info, typo and negation, demonstrating its capability in robustness.

7 Discussion

7.1 Major limitation

One major limitation of our challenging SRL model is that, our challenge dataset only has 50 test cases for testing 8 capabilities, which may not suffice the requirement of exhaustively, which is one of the key properties of challenge datasets. Even though we utlized various lexical resources, namely the Universal Proposition Databanks and PropBank, the test case creation method is still highly manual, not implementing automatic functions to generate test instances. In future experiment, we will attempt to adapt more automatic methods such as using the CheckList package[8].

7.2 BERT-based vs. Bi-LSTM

In terms of failure rate on our challenge dataset, the BERT-based SRL model slightly outperform Bi-LSTM model. In each capability test, the failure rate of BERT-based model is either equal with or lower than the Bi-LSTM model. However, given the size and quality limitation of our challenge dataset, we are not able to draw a general conclusion that to what extent these models are able to capture these capabilities, or the former model learnt better than the later. Rather, we argue that the main purpose and meaning of this experiment provides us a basic outline of SRL challenge dataset framework, and an opportunity to conduct an initial exploration into capabilities.

7.3 Capabilities

Through capability tests, here we provide a potential interpretation of the failure rates of capabilities: those connected with clear patterns tend to have lower failure rates, whereas those with more complex syntactic structures tend to fail more.
For example, both NER and Voice have a failure rate of 0. For NER, Named entities are all nominal words that usually follow a proposition or a verb, and for voice, passive voice usually has a fixed syntactic structure like 'object + verb (passive form) + by + subject'. In addition, the short

[8] https://github.com/marcotcr/checklist

subordinate clauses used in our challenge dataset also have fairly simple syntactic structures. In the lexical aspect, the low failure rate of causative inchoative alternation tests may due to the fact that the number of such verb pairs is small in English vocabulary, thus there is less variance in terms of lexis for this capability. On the contrary, capabilities such as identifying high position arguments involve many different syntactic variances in each argument position.

7.4 Model improvement

We propose the below directions for future model improvement based on our challenge dataset experiment. First and foremost, delve into more potential capabilities for SRL models, and implement more complex and sophisticated capability tests. We may also implement full argument span evaluation in order to get a comprehensive understanding of models' prediction on argument constituents. Second, to improve the performance of POS, we may add more annotated instances that contain noun predicates in the supervised stage of model training. Third, to improve the capability of high position argument identification, we may need to review the overlaps between high position arguments and adjunct arguments annotations in the training data.

8 Future work

Belinkov (2019) reviews analysis methods in NLP, categorizes them according to prominent research trends and also point out limitations and directions. The key factors that discussed in machine learning interpretability are accountability, trust, fairness, safety and reliability. For future analysis, below questions need to be emphasised: what linguistic info is capture in neural networks, which phenomena they are successful at capturing and where they fail.

The paper introduces the below approaches and their limitations:

Auxiliary prediction task It is also be referred to 'diagnostic classifiers' and 'probing tasks'. The typical approach is that, for a neural network model, after it is trained on a main task, we then use it for another classification task (such as POS). The performance of this auxiliary task is used as a proxy for evaluating the quality of the generated representation of the original model. The results demonstrate the kind of insights that the classification analysis may lead to, especially when comparing different models or model components.

However, the use of classification tasks lack both theoretical foundation and a better empirical consideration of the link between the auxiliary tasks and the original tasks.

Different components to be explored in Neural Network: we may also want to investigate different components of the neural network, including word embeddings, RNN hidden states or gate activation, sentence embeddings, and attention weights in sequence-to-sequence (seq2seq) models. Visualization is a useful tool to help us investigate these components. Early work visualized hidden unit activation in RNNs trained on an artificial language modeling task, while much recent work has focused on visualizing activation on specific examples in modern neural networks for language.

The current limitation of these is, not enough work has been done on explaining prediction of neural network despite various visualization tools are made available to use by NLP researchers.

Challenge sets In addition, an alternative evaluation framework consists of challenge sets, also known as test suites, which has been introduced and applied in our experiment. Nevertheless, now most of analysis work of the challenge datasets is focused on English language and tasks such as NLI and MT tasks.

9 Conclusion

In this challenging SRL models experiment, we first discuss the bias in the standard evaluation method, and introduce an evaluation methodology CHECKLIST. For semantic role labeling task, we present its background, propose 9 related capabilities and design a challenge dataset, including 50 test cases of MFT and INV test types, to evaluate 8 of them on two SRL models: the BERT-based and the Bi-LSTM. Then we analyse test results of each capability and discuss them from the aspects of models' performance and the limitations of our challenge dataset, provide a potential explanation of the difficulties in learning each capability and future model improvement approaches, including implementing more exhausted challenge capability test cases. Finally, we discuss the trends and limitations of analysis methods by reviewing relevant literature.

References

B Beizer. 1995. *Black-box testing: techniques for functional testing of software and systems*. John Wiley Sons, Inc.

Glass J. Belinkov, Y. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Dan Jurafsky and James Martin. 2019. *Speech and Language Processing*. Prentice Hall.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*, volume 3.

C. Piñón. 2001. A finer look at the causative-inchoative alternation. *In Semantics and linguistic theory*.

Roth D. Yih W. T. Punyakanok, V. 2008. The importance of syntactic parsing and inference in semantic role. *Computational Linguistics*, 34(2):257–287.

Wu T. Guestrin C. Singh S. Ribeiro, M. T. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. pages 885–895.