Take Home Exam

Natalia Khaidanova
2778662
GitHub: https://github.com/NataliaKhaidanova/SRL_challenge_sets

Abstract

In this study, nine SRL capabilities of two publicly available pre-trained SRL systems, i.e., structured-prediction-srl-bert (allenBERT) and structured-prediction-srl (allenBiLSTM), are evaluated using the idea of challenge sets presented in the CheckList library. The experiments revealed interesting results, such as an extremely poor performance of both models on passive-voice sentence constructions, that proved the advantage of challenge sets over regular evaluation methods.

1 Introduction

The majority of benchmark datasets used in NLP come from text corpora that represent a natural frequency distribution of language phenomena. While useful in practice for evaluating system performance in the average case, such datasets may fail to capture a wide range of phenomena. Training a system on such a biased dataset can result in systematically incorrect predictions. To mitigate this bias, behavioral, i.e., black-box testing can be implemented. Such an approach involves evaluating a model's performance by validating its input and output, without modifying its internal structure. This enables comparison between different models, without any intervention in their internal mechanisms. Challenge sets are one form of behavioral testing approach. The primary goal of such test sets is to evaluate a model on its ability to handle specific linguistic phenomena.

CheckList (Ribeiro et al., 2020) is an NLP library that is designed specifically for this purpose. The usefulness of CheckList was demonstrated on the three NLP tasks, i.e., *sentiment analysis* which is the task of determining the sentiment or emotion conveyed in a piece of text, *duplicate question detection* which involves identifying whether a given question is a duplicate of another question in a corpus, and *machine comprehension* which implies teaching machines to understand text passages and answer questions based on them.

CheckList suggests evaluating a model's performance using three test types, i.e., *Minimum Functionality test* (MFT), *Invariance test* (INV), and *Directional Expectation test* (DIR).

The purpose of MFT is to determine whether a model's performance meets minimum requirements. The minimum requirements are the basic capabilities, e.g. understanding the order of events, that a specific model must have to be considered operational. MFT requires a set of simple labeled examples within a certain capability that is being tested. Such a test set is particularly useful for uncovering severe bugs by detecting the instances when the model uses shortcuts without actually mastering the capability.

INV and DIR are used to assess whether a model remains consistent across different examples. Both these tests are perturbation-based. The perturbation can involve changing people's and location names to evaluate whether a model understands named entities at an appropriate level, etc. The difference between the two approaches is that INV assumes changing the input of the model and preserving the label it needs to predict, while DIR expects a change in the label.

To provide a systematic way of assessing different models across various capabilities, CheckList uses a matrix that represents a set of tests as rows and a set of models that are being evaluated as columns. The matrix can be customized and adapted to suit the needs of evaluation.

Test cases can be created both from scratch and with the use of perturbation. Creating test examples from scratch demands a lot of imagination and hard work, resulting in high-standard but low-coverage test instances. In contrast, perturbation functions are able to generate multiple examples at once. To facilitate the latter, CheckList pro-

vides different abstractions, e.g., templates, lexicons, general-purpose perturbations, visualizations, and context-aware suggestions. It is also possible to use RoBERTa, a masked language model, to get suggestions for word substitutions which the user can filter into positive, negative, and neutral substitution lists and later reuse across multiple tests.

In the aforementioned paper, the creators of CheckList provided a usage example that involves testing whether different models have the necessary vocabulary and whether they can appropriately handle the impact of words with different parts of speech. For this evaluation, multiple test sets were generated, each converting different types of examples. For MFT, short sentences with neutral adjectives and nouns were created for the models to predict their sentiment. For INV, some neutral words in a sentence were replaced with some other neutral words. The models' task was not to get confused and predict the same label they had predicted before the alternations. For DIR, negative sentences were added after the existing test examples. The models should have recognised the possible change in the sentiment and output a new label. Such simple tests revealed a failure rate of up to 94.6% in some of the models uncovering significant bugs that needed to be fixed.

The approach used in CheckList proved to be beneficial even for testing large public-facing systems, e.g., the general purpose sentiment analysis model by Microsoft that had already undergone multiple tests. The developers team discovered many previously unknown bugs by testing the model's benchmark capabilities more thoroughly and systematically or by evaluating the capabilities that had not previously been considered.

2 Background

Semantic Role Labeling (SRL) is a shallow semantic parsing task, in which for each predicate in a sentence, the goal is to identify all arguments that fill a semantic role and to determine said roles, e.g., Agent, Patient, Instrument, etc. and their adjuncts, e.g., Locative, Temporal, Manner, etc. (Punyakanok et al., 2008), i.e., the goal is to find out who did what to whom, where and when.

In CheckList, SRL capabilities of the models were assessed for the three tasks it was evaluated on, i.e., *sentiment analysis*, *duplicate question detection*, and *machine comprehension*. All test examples were created based on the MFT test type.

The SRL capabilities important for sentiment analysis included determining whether a model understands that a sentiment expressed by the author of the text is more important that the other Agents' opinions, e.g., the sentence *Some people hate you, but I think you are exceptional.* expresses a positive sentiment, even though the first independent clause exhibits a negative connotation. The other two test sets involved evaluating a model's performance on the examples with the question + Yes/No answer structure, e.g., *Do I think that is an awkward customer service? Yes.* expresses negative emotions, whereas the sentence *Do I think this company is bad? No.* contains either a positive or neutral sentiment.

The test sets created to evaluate a system for duplicate question detection included five SRL capabilities. The first three capabilities involved understanding the relevance of the word order in comparative structures and symmetric relations and its relevance in asymmetric relations, e.g., the sentences in the following pairs *Are tigers heavier than insects? What is heavier, insects or tigers?* and *Is Nicole related to Heather? Is Heather related to Nicole?* are duplicates of each other, whereas *Is Sean hurting Ethan? Is Ethan hurting Sean?* differ in meaning. The other two SRL capabilities included detecting the presence or absence of a semantic change in active/passive sentence constructions, e.g., the sentences *Does Anna love Benjamin? Is Benjamin loved by Anna?* duplicate each other, while *Does Danielle support Alyssa? Is Danielle supported by Alyssa?* do not.

The SRL capabilities used to assess the performance of a model in machine comprehension were compiled into two test sets aimed at testing a system's ability to distinguish between subjects and objects in a sentence. The first test set included simple sentence examples, e.g., *C: Richard bothers Elizabeth. Q: Who is bothered? A: 1) Elizabeth 2) Richard* where Elizabeth is the correct answer. The second test set comprised more complex cases, e.g., *C: Jose hates Lisa. Kevin is hated by Lisa. Q: Who hates Kevin? A: 1) Lisa 2) Jose* with Lisa being the correct answer to the question.

Since CheckList did not test the SRL capabilities of the models for the SRL task but for the other tasks mentioned earlier, not all of the aforementioned capabilities are relevant to the SRL task itself.

For instance, the SRL capability of a model to determine a core sentiment of a sentence expressed by the author of the text is essential for sentiment analysis but not for SRL as it does not imply difficulty in identifying semantic roles.

Similarly, the SRL capabilities of a model to identify a sentiment of the structure based on the question + Yes/No answer are, for the same reason, not important for the SRL task itself but rather for sentiment or emotion detection.

The SRL capabilities of a system to distinguish between subjects and objects in a sentence are, on the contrary, essential not only for machine comprehension but for the SRL task as well. However, most of the SRL models are likely to master these capabilities as simple subject-verb-object constructions are expected to compose a vast part of their training examples.

The tests that involve evaluating a model's capabilities to handle a different word order are generally meant to capture predicate meaning. Subsequently, understanding the meaning of the predicate is a core capability of an SRL system as semantic roles are often determined by its meaning. For instance, in the sentence *John broke the window.*, the predicate *broke* implies that John is the Agent (ARG0) who performed the action and *the window* is the Theme (ARG1). However, the same predicate can be used with different roles depending on the context, e.g., in the sentence *The ball broke the window.*, the predicate *broke* suggests that *The ball* is the Instrument (ARG2) and *the window* is ARG1.

In case of a caused-motion argument alternation where ARG0 is absent, a model needs to have the capability to distinguish such constructions from regular sentence structures, e.g., to distinguish between *The window broke.*, where *The window* is ARG1, and *John left.*, where *John* is ARG0.

Likewise, sentences with a benefactive argument alternation may appear challenging for an SRL system as a verb can take either a direct object or a prepositional phrase that indicates the recipient of the action. Nevertheless, an SRL system should be able to identify the alternation in the sentence of the type *She baked her sister a cake* and assign *her sister* the role of ARG2 instead of ARG1.

In addition, in passive sentence constructions, the subject and object interchange, if compared to active voice structures, e.g., in the sentence *The car was repaired by the mechanic.*, *The car* is the object and has the role of ARG1, whereas *the mechanic* is the subject and holds the role of ARG0. An SRL system should be able to distinguish between active and passive sentence examples and take into account the changes in semantic roles.

Dealing with elliptical structures may also be challenging for an SRL model. Yet, it is one of the core capabilities it should have. An SRL system should be able to identify the left-out elements and assign correct roles to the remaining arguments, e.g., given the sentence *John likes coffee, and Mary tea.*, it should find the predicate missing in the second independent clause, *likes*, and assign the roles of ARG0 and ARG1 to *Mary* and *tea*, respectively. Similarly, in the case of the left-out Theme, such as in the sentence *He speaks French, and so does his wife.*, a model should be able to identify that the sentence is compiled of two independent clauses and to assign correct roles to the arguments, i.e., label *He* and *his wife* as ARG0 and *French* as ARG1.

Additionally, understanding ambiguous sentences is another capability an SRL system should possess. The ambiguity is often caused by polysemantic predicates. For instance, in the sentence *I saw a kid with a cat.*, the verb *saw* can be viewed by an SRL system as an act of seeing or sawing. In the first case, *a cat* would be ARG1, whereas in the second case, it would take the role of ARG2. A model should estimate the probability and assign *a cat* the label with the highest probability, which in this case is ARG1.

Finally, an SRL system should have the capability to understand standard idiomatic expressions, e.g., *to kick the bucket* which means *to die*. In the given example, *the bucket* does not have a specific semantic role as it appears in the idiomatic context.

2.1 Lexical Resources

As most of the aforementioned capabilities of an SRL system are based on the differences in predicate senses, it is beneficial to use external lexical resources, e.g., VerbNet (Palmer et al., 2005) or PropBank (Babko-Malaya et al., 2004) to facilitate the creation of challenge test sets.

VerbNet is a hierarchical network of verb senses, organized into classes based on shared syntactic and semantic properties. Each sense of a verb in VerbNet is associated with a set of thematic roles, which describe the semantic relationships between the verb and its arguments. PropBank, on the other hand, is a corpus-based resource that provides anno-

SRL Capability	Test Type and Description	Example Test Cases & Expected Behavior
Argument alternation	**MFT:** Instrument+Theme	The ball broke the window. [*ARG2* The ball] broke [*ARG1* the window].
	MFT: Caused-motion	The window broke. [*ARG1* The window] broke.
	MFT: Benefactive	She baked her sister a cake. [*ARG0]* She] baked [*ARG2* her sister] [*ARG1* a cake].
	MFT: Passive voice	The car was repaired by the mechanic. [*ARG1* The car] was repaired by [*ARG0* the mechanic].
Elliptical sentences	**MFT:** Left-out predicate	John likes coffee, and Mary tea. [*ARG0* John] likes [*ARG1* coffee] and [*ARG0* Mary][*ARG1* tea].
	MFT: Left-out Theme	He speaks French, and so does his wife. [*ARG0* He] speaks [*ARG1* French], and so does [*ARG0* his wife].
Ambiguity	**MFT:** Polysemy	I saw a kid with a cat. [*ARG0* I] saw [*ARG1* a kid with a cat].
		He saw a man with binoculars. [*ARG0* He] saw [a *ARG1* man] [*ARG2* with binoculars].
	MFT: ARG0 idiomatic expressions	He hit the books. [*ARG0* He] hit the books.
	MFT: ARG1 idiomatic expressions	She kicked the bucket. [*ARG1* She] kicked the bucket.

Table 1: A selection of tests for SRL.

tations for a large corpus of English text. It contains more than 3600 verb frames and 5050 framesets that refer to specific verb meanings (Pazienza et al.). Each frameset has its own set of semantic roles that need to be assigned to the arguments depending on the predicate sense. Searching for identical sets of semantic roles can help find verbs that form similar syntactic structures and thus build the test sets that would help evaluate certain capabilities of an SRL system.

The use of other lexical resources like WordNet (Miller, 1992) can also facilitate the creation of challenge sets by providing synsets, i.e., synonymous substitutions for certain words. Therefore, it can be used to easily extend manually created examples and increase the lexical coverage of test instances.

3 Challenging SRL

A range of nine challenge test sets aimed at assessing the capabilities of an SRL system mentioned in Section 2 is listed in Table 1. The table groups the tests into three categories based on the capability type, i.e., *Argument alternation* (ability to deal with different argument order), *Elliptical sentences* (linking arguments to the left-out predicate or Theme), and *Ambiguity* (understanding the primary meaning of the sentence). Test examples are provided along with the expected system behavior.

4 Creating a Challenge Dataset

Nine challenge test sets were created either manually or semi-automatically. The semi-automatic approach involved manually creating a small set of examples and extending it using the fill-mask pipeline [1] of the Transformers library.

A test set containing Instrument+Theme examples was created based on the 18 manually produced instances which were subsequently multiplied by fill-masking the predicate. This enabled increasing the number of test instances to 104 examples. However, after manual verification, the number of examples was reduced to 45. It was determined not to fill-mask the subject as it would result in a great number of incorrect ARG0 substitutions. Therefore, it should be taken into account that the 45 test examples might not be representative enough of a model's performance.

To investigate a system's response to caused-motion argument alternation, a set of 17 manually crafted examples was created. In this case, it was decided not to use the fill-mask pipeline since, as it was established practically, the semi-automatic approach resulted in a substantial number of incorrect substitutions. Due to time constraints, it did not appear feasible to manually verify more than 300 machine-generated instances. Therefore, it was determined to leave the set of caused-motion examples small but of high quality.

Examples of benefactive argument alternation

[1] https://huggingface.co/tasks/fill-mask

were produced based on the nine manually created instances. The fill-mask pipeline was used to get substitutions for the predicate and direct object resulting in 232 test instances which were later manually reduced to 177 by removing incorrect generations. It was determined not to fill-mask the subject and indirect object as substituting four tokens in a sentence would result in more than 2000 test instances. After the reduction, the gold labels for the recipient were manually edited. It appeared to be necessary since some verbs, e.g., *to offer* or *to buy* required a different label for this argument rather than ARG2, as stated in PropBank.

To create passive voice examples, a set of 16 manually produced instances was extended with the fill-masked subject and the main verb of the predicate. Such a methodology enabled increasing the number of test examples to 406. The quality of all instances was verified manually. Fill-masking the prepositional phrase was not conducted due to the subsequent substantial increase in the number of test instances.

To evaluate a system's capability to identify semantic roles in the elliptical sentences with the left-out predicate, 12 test examples were created manually and extended by fill-masking the predicate which, after manual investigation, resulted in 39 test instances. Fill-masking the direct objects was not performed as masking two tokens simultaneously was not supported by the Transformers fill-mask pipeline. However, filling in the substitutions for the two tokens separately would result in an immense number of incorrect fill-ins as the two direct objects would not correspond in the vast majority of the generated examples.

The test examples of elliptical sentences with the left-out Theme were generated based on the manually created nine instances. Fill-masking the predicate and the direct object resulted in a substantial increase in the number of test cases which were manually inspected for any grammatical errors and subsequently reduced to 197.

To test the capability of an SRL system to deal with polysemantic expressions that capture syntactic ambiguity, a test set of two examples, i.e., *I saw a kid with a cat.* and *He saw a man with binoculars.* was created. In both cases, the prepositional phrase may be assigned the role of ARG2. However, the probability of being ARG2 differs with the phrase *with a cat* having a smaller probability than the expression *with binoculars*. Therefore, in the first expression, *cat* was given the role of ARG1, whereas in the latter sentence, *binoculars* was labeled as ARG2.

Finally, 35 examples of idiomatic expressions were created manually to evaluate a model's capability to distinguish sentences with indirect meaning from all other instances. The gold labels were determined based on the PropBank framesets corresponding to the primary meaning of the expression. Subsequently, the test examples were divided into two sets, one with 23 examples of the subject being ARG0 and one with 9 examples of it being ARG1. For instance, the expression *to hit the books* means *to study*. As the corresponding PropBank frameset suggests two semantic roles, student (ARG0) and subject (ARG1), that can be assigned to the syntactic subject and object of the sentence with such a predicate, in the example *He hit the books.*, *He* was labeled as ARG0. Another example is the aforementioned idiomatic expression *to kick the bucket* which means *to die*. Its corresponding frameset contains two possible semantic roles, the deceased (ARG1) and cognate object (ARG2). Therefore, in the sentence *She kicked the bucket*, *She* was given the role of ARG1.

5 Evaluating Models

The aforementioned SRL capabilities were evaluated for the two publicly available models, i.e., structured-prediction-srl-bert (allenBERT) [2] (Shi and Lin, 2019) which is a BERT-based model fine-funed for SRL by AllenNLP and structured-prediction-srl (allenBiLSTM) [3] (He et al., 2017) which is based on a deep BiLSTM sequence prediction model and is available at the same platform.

It is important to note that both these models produce SRL labels for full constituents. Therefore, the predicted labels are reduced to ARGs to suit the needs of evaluation.

Even though the aforementioned models represent the state of the art in SRL, they are not expected to perform well on the challenge sets.

For instance, in the Instrument+Theme test examples, they are likely to confuse the majority of ARG2 instances for ARG0. Making the correct prediction requires such instances to constitute a

[2] https://github.com/allenai/allennlp-models/blob/main/allennlp_models/modelcards/structured-prediction-srl-bert.json

[3] https://github.com/allenai/allennlp-models/blob/main/allennlp_models/modelcards/structured-prediction-srl.json

Test Type and Description	Failure Rate % allenBERT	Failure Rate % allenBiLSTM	Nr. Correct allenBERT	Nr. Correct allenBiLSTM	Total
MFT: Instrument+Theme	23.0	31.9	14	3	45
MFT: Caused-motion	2.9	2.9	16	16	17
MFT: Benefactive	1.6	7.2	168	147	177
MFT: Passive voice	50.0	50.0	0	0	406
MFT: Left-out predicate	32.7	31.7	0	0	39
MFT: Left-out Theme	20.2	20.3	0	0	197
MFT: Polysemy	13.3	13.3	1	1	2
MFT: ARG0 idiomatic expressions	40.0	40.0	0	0	23
MFT: ARG1 idiomatic expressions	60.0	60.0	0	0	9

Table 2: Failure rate and the number of fully correct predictions out of the total number of test instances of allenBERT and allenBiLSTM on the SRL challenge sets.

sufficient part of these models' training examples which appears to be unlikely. Therefore, there is a high chance that the models would not be able to distinguish such test cases from similar examples with the subject being ARG0.

The caused-motion argument alternation examples may be easier for these models to handle. There is a possibility that they would incorrectly label ARG1 as ARG0. However, sentences with ARG0 typically form a subject-verb-object structure. The evaluated models should have the knowledge that if a sentence has only two parts, it is probable that it contains caused-motion argument alternation.

Resolving benefactive argument alternation should not appear problematic as neural networks should have captured the meanings of certain verbs, e.g., *give* or *land* that typically require a recipient and a subject that is given or lent, during their training. Nevertheless, the argument order in such sentence constructions differs from the standard argument sequence. Therefore, it is still important to evaluate the models' performance on such test examples.

Hypothetically, labeling arguments in simple passive-voice constructions should not be challenging for state-of-the-art SRL systems either. However, real-world experience shows that such cases are rarely resolved correctly. Therefore, there is a high chance that the aforementioned models will switch places for ARG1 and ARG0 as in active-voice sentence constructions.

Elliptical sentences with the left-out predicate, as well as the sentences with the left-out Theme, are likely to be one of the least mastered capabilities of any SRL system. Such cases require an individual approach during a model's training. Therefore, all types of elliptical sentences are expected to be labeled incorrectly by the models being evaluated.

In the Polysemy test set, the first example is expected to be labeled correctly, whereas the second example may get predictions that would contradict the gold labels. However, not labeling *with binoculars* as ARG2 should not be viewed as a serious error as this sentence may have different interpretations. Nevertheless, such misclassification may indicate that a model is likely to make the same mistake in cases that do not support different interpretations.

Similarly, none of the examples of idiomatic expressions is expected to be labeled fully correctly. In the case of ARG0 idiomatic expressions, the models are likely to assign an additional label to the phrasal verb as they may mistake it for the verb and the direct object. However, there is a high chance that they would label the subject correctly as the structure of ARG0 idiomatic expressions resembles regular subject-verb-object constructions. ARG1 idiomatic expressions are, on the contrary, expected to be labeled fully incorrectly as not only do not they contain the direct object but also their subject has the role of ARG1.

Overall, the BERT- and BiLSTM-based models are expected to perform the worst on the following test sets: Instrument+Theme, Left-out predicate, Left-out Theme, and ARG1 idiomatic expressions.

6 Results

As SRL is a sequence labeling task, the failure rate is calculated based on the predictions of all labels in a sequence. However, due to the difference in the number of tokens per instance in each test set,

the failure rate may not accurately reflect the performance of the models. Therefore, the number of fully correct predictions, i.e., the instances where the model predicted all labels in a sequence correctly, are included in Table 2 together with the failure rate and total number of test examples per capability.

We can see that the allenBERT model significantly outperforms allenBiLSTM in the Instrument+Theme capability. As determined from the error analysis, allenBiLSTM has only correctly predicted the instances with *bullet* being the subject followed by one of the following predicates: *struck, hit* or *pierced*. The majority of the instances being labeled correctly by allenBERT also include the verbs *pierced, hit* and *struck* but also the verbs *broke* and *killed*. Surprisingly, allenBERT has labeled *knife* and *scissors* as ARG3 in some of the examples where they are followed by the predicate *sliced* and *cut*, respectively. This should not be viewed as a serious error as Instrument can also have the role of ARG3. Nevertheless, interestingly enough, it has labeled *knife* neither as ARG2 nor ARG3 in the example *The knife cut the bread.* Overall, we can notice that the models are able to recognize benefactive argument alternation only when certain words are present in a sequence.

As expected, both allenBERT and allenBiLSTM have assigned correct labels to almost all of the instances of caused-motion argument alternation. The only test example the models failed to label correctly is *The balloon burst.* In both cases, the cause of the failure was the predicate *burst* which the models failed to identify as an event and, therefore, were not able to assign any labels to the sequence.

Benefactive argument alternation did not appear to be a very challenging task either. Both allenBERT and allenBiLSTM resolved almost all of the test instances correctly. However, in this case, the performance of the models differs significantly. For allenBERT, the examples with the predicate *cooked* seemed to be the most difficult as they compose three out of the total nine mistakes. The other mistakes made by this model originate from mislabeling the sequences with the following predicates: *baked* (two mistakes), *poured* (two mistakes), *ordered* (one mistake), and *read* (one mistake). Remarkably, in the sentence *She baked her grandmother a pie.*, allenBERT labeled *her grandmother* as ARG4, whereas in the example *They baked their guests a cake.*, *their guests* was assigned the role of ARG3. For allenBiLSTM, the instances with the predicate *bought* appeared to be the most challenging as they resulted in ten additional mistakes made by the model. Interestingly enough, allenBiLSTM did not make the same errors as allenBERT. The only instance both of these models labeled incorrectly is *She ordered her friend a drink.* where they assigned the role of ARG1 to *her friend* and the role of ARG2 to *a drink*.

None of the 406 passive-voice test examples was labeled correctly which proves the fact that passive constructions are difficult to label even for state-of-the-art SRL systems. Both models struggled to even recognize passive voice and identify an event. Subsequently, all tokens in the vast majority of the cases were labeled as O. When the models did label the given sequence with anything apart from O, they mistook a form of the verb *to be* for the event which potentially led to misclassifying the main verb and the subject for ARG2.

Similarly, none of the elliptical sentence examples with the left-out predicate or left-out Theme was resolved correctly by any of the models. In the case of the left-out predicted, allenBiLSTM was able to predict correctly slightly more labels than allenBERT. For the left-out Theme, the results are the opposite. However, in both cases, the difference is scarce. Remarkably, in the sentences with the missing second predicate, both models typically label all tokens in the second independent clause as either ARG1 or O. In the examples with the left-out Theme, both allenBERT and allenBiLSTM assign O to all tokens following the first independent clause.

In the examples of polysemantic expressions, both models predicted the labels for the first sequence, i.e., *I saw a kid with a cat.* correctly, whereas none of the models was able to assign correct semantic roles for the arguments in the second example, i.e., *He saw a man with binoculars.* AllenBERT mistook *with binoculars* for ARGM-MNR, i.e., the manner in which the action is performed, whereas allenBiLSTM assigned this phrase the role of ARG1.

Finally, none of the idiomatic expressions was resolved correctly by any of the models being evaluated. Both of them were not able to distinguish between ARG0 and ARG1 idiomatic expressions labeling all syntactic subjects as ARG0. Moreover, in all cases, the phrasal verb was mistaken for the verb and the direct object, as it was expected.

Overall, allenBERT showed better performance compared to allenBiLSTM outperforming it in all of the aforementioned SRL capabilities, except for the elliptical sentences with the left-out predicate.

7 Discussion

In this research, we tested only nine SRL capabilities. Even though the experiments revealed interesting and sometimes unexpected results, more capabilities should be evaluated to make precise conclusions about the quality of the models. Moreover, some of the test sets with fewer than 100 instances lacked diversity in their examples. This means that the number of examples should, ideally, exceed the given number. Extending the representation is, nevertheless, difficult in the majority of cases. For instance, the gold labels for benefactive argument alternation should be verified manually as it was determined that even similar predicates may have different labels for the recipient according to PropBank. In the sentence with the predicate *gave*, the recipient should be assigned the role of ARG2, whereas in the instance with the predicate *offered*, it should be labeled as ARG3. Furthermore, some verbs, e.g., *cook, email, pour, text, bake* and *gift* either do not have a PropBank frame or do not have a recipient as a possible argument. All of this complicates the creation of high-quality test sets and, subsequently, the evaluation of the models.

8 Future work

Challenge sets are especially useful for the tasks like domain adaptation as they show the direction in which domain adaptation should be performed. For instance, if we have a small amount of labeled medical data and a larger corpus of unlabeled data, we can create specific domain-adapted challenge sets and evaluate the pre-trained SRL model later to be fine-tuned for the medical domain. The performance of the model will indicate the aspects to focus on when performing domain adaptation.

To do that, it is important to first develop a lexicon and a set of semantic roles that are relevant to the medical domain. This can involve analyzing the available data, consulting domain experts, or reviewing the corresponding literature. The next step would be selecting a representative subset of the unlabeled data. This subset should be big enough to cover a wide range of domain-specific variations. Combining it with the labeled examples would create a test set for estimating the current performance of the model and further refining the domain-specific ontology. Once we have a complete understanding of the capabilities a medical SRL system should possess, we can start creating challenge sets, each representing the necessary property of the model.

It can be assumed that the majority of the MFT test sets would be difficult to generate as it would likely require a deep knowledge of the medical field. Building perturbation-based INV and DIR test instances may, in this case, be a better choice. The INV challenge tests may involve fill-masking named entities, substituting terminology for the WordNet synsets, or introducing typos in the instances present in the data. DIR examples may include converting active sentences to passive ones and other operations. The full list of the capabilities to test can only be established once the precise medical ontology is developed.

The created challenge test sets can be used to fine-tune the pre-trained SRL system for the medical domain.

9 Conclusion

The experiments conducted in this research demonstrate that creating challenge sets is an effective domain-independent and model-agnostic method that can accurately estimate the specific capabilities of any SRL system. In this study, two pre-trained SRL models, i.e., allenBERT and allenBiLSTM, were assessed with the help of nine challenge sets. The sets were divided into three categories according to the core SRL capability they tested, which included argument alternation, elliptical sentences, and ambiguity.

The results of the experiments revealed that the performance of the models varied significantly across the different challenge sets. The models were able to correctly resolve none of the elliptical sentence examples, which comprised two test sets: one with a missing second predicate and one with the left-out Theme. This finding suggests that elliptical sentences are the most challenging core SRL capability for both models. On the other hand, the models performed slightly better on the ambiguous test examples, which included polysemantic sentences, ARG0 and ARG1 idiomatic expressions. Additionally, the experiments on argument alternation challenge sets yielded interesting results. Passive-voice constructions were found to be exceptionally challenging for both models, while

the Instrument+Theme test instances uncovered a significant difference in the models' performance, with allenBiLSTM showing much worse results than allenBERT. Interestingly enough, the caused-motion and benefactive argument alternation tests did not appear to pose significant challenges for either model, with both models resolving the majority of instances correctly.

These results suggest that SRL models may have varying strengths and weaknesses in different SRL capabilities, highlighting the importance of developing diverse challenge sets to accurately evaluate their performance.

References

Olga Babko-Malaya, Martha Palmer, Nianwen Xue, Aravind Joshi, and Seth Kulick. 2004. Proposition bank II: Delving deeper. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 17–23.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

George A. Miller. 1992. Wordnet: A lexical database for english. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Maria Teresa Pazienza, Marco Pennacchiotti, and Fabio Massimo Zanzotto. Mixing WordNet, VerbNet and PropBank for studying verb relations.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.