

Challenging SRL Models

Csenge Szabo
2803781

1 Introduction

Bias is an inherent challenge in Natural Language Processing (NLP) systems, particularly those employed for predictions. Neural networks models might learn non-generalizable patterns closely from the training data, leading to poor generalization capabilities, overfitting and finding shortcuts when applied on unseen data. Models might also amplify undesirable biases observed during training, which can lead to predictions with systematic discrimination against certain demographic groups (Shah et al., 2020). Traditionally, the performance of NLP models is assessed using held-out datasets and evaluation scores. This approach can overestimate the real-world performance of systems because the test set inherits the same biases as the training set. In contrast with training-validation-test splits that cannot efficiently capture systematic shortcomings of a model, behavioral testing¹ involves carefully constructing test samples and observing the models' output labels on these samples. This approach allows for testing specific capabilities of models trained on different datasets without focusing on the models' architecture (Ribeiro et al., 2020).

Ribeiro et al. (2020) introduce CHECKLIST, a novel evaluation technique based on the principles of behavioral testing for comparing NLP models. The authors illustrate the utility of CHECKLIST by comparing state-of-the-art models on three NLP tasks: sentiment analysis, question pair detection and machine comprehension. Sentiment Analysis involves predicting the sentiment polarity of a text sequence with a scale or the labels *positive*, *negative* and *neutral* (Nakov et al., 2019). Question pair detection is a binary classification task to determine if two questions convey the same meaning (Shah et al., 2018). Machine comprehension relates to the ability of a system to understand, interpret and answer questions based on an input text (Rajpurkar et al., 2016). CHECKLISTs employ various test types targeting a selection of linguistic capabilities. By

analyzing the models' failure rates on these tests, researchers can gain a better understanding of the models' weaknesses and biases (Ribeiro et al., 2020).

CHECKLIST

Tests are structurally organised into a matrix structure, where rows represent specific linguistic capabilities and columns represent different test types, including Minimum Functionality test (MFT)², Invariance test (INV)³ and Directional Expectation test (DIR)⁴. The cells in the matrix display the model's failure rate on a tested capability with a specific test type. Some cells may be left empty if the capability-test combination was not suitable or applicable for testing the NLP task at hand (Ribeiro et al., 2020).

Test instances can be created in several ways. They can be manually crafted, which can be time-consuming and result in a dataset with limited size and coverage. An alternative is generating test instances and perturbations using predefined templates with placeholders, for example, *The {THING} is {COMP_ADJ} than the {THING}*. This method offers a larger and more diverse test set but still relies on the creator's creativity. Advanced techniques involve leveraging masked language models like RoBERTa for the automatic generation of test instances. Here, a sentence with a masked phrase can be input for the language model, which suggests a ranked set of potential fill-ins that can be filtered and used for various tests. Example input: *She certainly {mask} dinner*. RoBERTa output: (1) *enjoyed*, (2) *liked*, (3) *ate* (Ribeiro et al., 2020).

Task-specific CHECKLISTs

This section reflects on the CHECKLIST method by Ribeiro et al. (2020) for evaluating sentiment analysis models. CHECKLIST utilises a combination of MFT, INV and DIR tests to assess various capabilities of the models on a dataset of airline tweets.

²Simple sanity checks for testing specific model behavior.

³Creating perturbations in the input, without any expected change in the output label.

⁴Creating perturbations in the input, with expected change in the output label.

¹It is also referred to as "black-box testing" since the internal structure of neural network models is difficult to explore.

- *Vocab.+POS*: These tests assess the models' ability to recognise common sentiment-bearing words.
- *Robustness*: INV tests evaluate the models' sensitivity to added noise like URLs and tags without an expected change in sentiment prediction.
- *Named Entity Invariance*: These tests measure the models' ability to maintain consistent sentiment predictions despite changes in location and person entities within the input text.
- *Temporal Aspects*: These tests ensure sentiment predictions remain consistent even when the temporal context of the sentence is altered.
- *Negation*: There are extensive MFT tests to evaluate the models' capability in understanding the impact of negation cues on the sentiment polarity of the sentence, considering variations in their placement and the original sentiment of the sentence.
- *Semantic Role Labeling*: This aspect explores the ability to handle sentiment in question-answer forms and identify the author's sentiment within the text.

2 Background

Semantic Role Labeling (SRL)

Extracting semantic information from textual data is essential for several NLP tasks, including question-answering, information extraction, and machine translation. The aim of SRL is to determine the properties and participants of an event by answering the questions "who" did "what" to "whom", "when", "where", and "how". SRL is a complex sequence labeling task composed of multiple sub-tasks: predicate identification, predicate classification, argument identification and argument classification (Carreras and Màrquez, 2005). This paper focuses on the task of argument identification and classification, presuming that information about predicates is provided. The evaluated models were trained on the PropBank v1.0⁵ dataset for SRL in English.

Proposition refers to the combination of the governing predicate and its group of arguments and adjuncts, thus one sentence may be composed of multiple propositions depending on the number of predicates (Gildea and Jurafsky, 2002). Predicates are not limited to verbs but may also include nouns and adjectives. The examples below illustrate the task of identifying and classifying the semantic role of each argument,

separated by square brackets, that belong to the same predicate.

- (1) [ARG0 John] **opened** [ARG1 the door] [ARG2 with his foot].
- (2) [ARG1 The door] **opened**.

In this sentence the verb 'open.01' is the predicate, and the semantic roles of its arguments are agent (ARG0), patient (ARG1), and the instrument (ARG2) in the first sentence. In the second sentence, the role of the first constituent is not agent-like but rather patient-like. This example pair shows that a combination of syntactic, lexical and semantic information is essential for handling SRL. Table 1 displays the most frequent argument types and their labeling conventions in PropBank (Bonial et al., 2012).

Argument	Description
ARG0	agent
ARG1	patient
ARG2	instrument, benefactive, attribute
ARG3	starting point, benefactive, attribute
ARG3	ending point
ARGM-	modifier

Table 1: Argument labels in PropBank

Capabilities of an SRL system

An efficient SRL system should be able to correctly label the arguments in the input sequence regardless of syntactic, lexical or semantic variations and level of structural complexity. For instance, the system should have an understanding of the impact of active-passive alternation or causative-inchoative alternation on the syntactic positions of ARG0 and ARG1 in the input sentence.

Ribeiro et al. (2020) propose a selection of test types that could be adapted for SRL. MFTs could be employed to evaluate models' capability to identify specific argument types given a linguistic phenomenon. INVs could assess a system's robustness against syntactical changes that do not alter semantic roles, such as negation. DIRs could be applied to examine the system's response to syntactic or lexical changes that affect role assignment, such as the dative alternation. The systems' capabilities can be evaluated using the failure rate metric, indicating the ratio of test instances the model failed on for a given test type.

When creating test scenarios for SRL systems, several existing resources offering structured information on verb roles can be utilised. PropBank⁶ provides detailed information on Universal Thematic Roles verbs can take, VerbNet⁷ organises verbs into classes based

⁵<https://universalpropositions.github.io>

⁶<https://propbank.github.io>

⁷<https://verbs.colorado.edu/verb-index/vn3.3/>

on shared semantic and syntactic behavior, FrameNet⁸ provides information on semantic frames and their frame elements, and WordNet⁹ is a lexical database that could be used to find lexical variability and synonymy (Palmer et al., 2005). The listed resources offer structured verb role information that can enrich the creation of tests for different syntactic and lexical capabilities. Generative large language models, such as ChatGPT-4, can support researchers in creating diverse test instances using templates and examples.

3 Challenging SRL

This section describes the proposed syntactic and lexical capabilities and test types for evaluating three transformer-based SRL systems. For examples and information about sample size per test type, refer to Table 2.

3.1 Voice

Active-Passive Transformation

Transformation of voice significantly alters the argument structure of sentences. This change is crucial for understanding the dynamics between the agent of an action and the recipient of that action. The agent (*Martha*) and the recipient (*the newspaper*) exchange positions in active and passive voice transformations, as the examples show below (Greenbaum, 1996). An advanced SRL system should be able to accurately identify ARG0 and ARG1, irrespective of the sentence’s voice and complexity.

Example

1. Martha [ARG0] takes the newspaper [ARG1].
2. The newspaper [ARG1] was taken by Martha [ARG0].

Test types and test description

- **MFT1:** Ability to identify ARG0 and ARG1 in short sentences with active voice.
- **MFT2:** Ability to identify ARG0 and ARG1 in short sentences with passive voice.
- **MFT3:** Predicate disambiguation: ability to identify ARG0 and ARG1 in sentences with active/passive voice and two identical predicates.
- **INV1:** Ability to identify ARG0 and ARG1 in passive sentences with long-range dependencies.

- **INV2:** Robustness against typos in the predicate while correctly identifying ARG0 and ARG1 in passive sentences.
- **DIR1:** Identification of the shift from ARG0 to ARG2 in passive sentences due to changes in prepositions (*for Martha* and *by Martha*).

3.2 Transitivity Alternation Patterns

Causative/Inchoative Alternation

Transitive verbs that are part of this alternation pattern express change of state or change of position, and they can be in both causative and inchoative sentence structures. Causative events express focus on the agent causing an action to happen, while in inchoative events emphasize the action and the agent is not disclosed. Levin (1993) categorises verbs into groups based on their common characteristics of syntactic and semantic behavior. The predicates in the created test instances belong to the verb class *break-45.1*¹⁰, which showcase this alternation pattern. To make the tests with causative sentences more challenging, I am alternating between volitional, non-volitional and instrument agents in the test instances, such as *sculptor*, *erosion*, *blades*.

Example

1. Causative: He [ARG0] broke the cup [ARG1].
2. Inchoative: The cup [ARG1] broke.

Test types and test description

- **MFT1:** Ability to identify ARG0 and ARG1 in short causative sentences using volitional, non-volitional and instrument agents.
- **MFT2:** Ability to identify ARG1 in short inchoative sentences.
- **INV1:** Ability to identify ARG0 and ARG1 causative sentences with long-range dependencies using volitional, non-volitional and instrument agents.
- **INV2:** Ability to identify ARG1 inchoative sentences with long-range dependencies.
- **INV3:** Measure robustness with negated inchoative sentences, ability to identify ARG1 inchoative sentences.

Preposition Drop Alternation

Transitive verbs that are part of this alternation pattern can be either part of an intransitive structure with a

⁸<http://framenet.icsi.berkeley.edu>

⁹<http://wordnetweb.princeton.edu/perl/webwn>

¹⁰<https://verbs.colorado.edu/verb-index/vn/break-45.1.php>

prepositional phrase, or in a transitive structure, where the preposition itself is "dropped" (Levin, 1993). The predicates in the created test instances belong to the verb class *run-51.3.2*¹¹.

Example

1. He [ARG0] swam through the English Channel [ARG1] last year.
2. He [ARG0] swam the English Channel [ARG1] last year.

Test types and test description

- **MFT1:** Ability to identify ARG0 and ARG1 intransitive sentences where ARG1 is part of a prepositional phrase.
- **MFT2:** Ability to identify ARG0 and ARG1 transitive sentences where the preposition is "dropped".

3.3 Alternations Involving Arguments in the VP

Dative Alternation

Verbs that are part of this alternation pattern can create double object constructions or prepositional indirect object constructions. In the former, the first NP is the recipient (*dative case*) and the second one is the theme. In the latter, the first NP is the theme followed by a preposition (*to*) and the recipient (Levin, 1993). This capability tests if the model is able to handle changes in syntactic structure without changing the semantic role labels. The predicates in the created test instances belong to the verb class *give-13.1*¹², which are part of this alternation pattern.

Example

1. Double object construction: Jack [ARG0] gave his brother [ARG2] a present [ARG1].
2. Prepositional indirect object construction: Jack [ARG0] gave a present [ARG1] to his brother [ARG2].

Test types and test description

- **MFT1:** Ability to identify ARG0, ARG1 and ARG2 in sentences with prepositional indirect object construction.
- **MFT2:** Ability to identify ARG0, ARG1 and ARG2 in sentences with double object construction.

- **INV1:** Ability to identify ARG0, ARG1 and ARG2 in sentences with prepositional indirect object construction and long-range dependencies.
- **INV2:** Ability to identify ARG0, ARG1 and ARG2 in sentences with double object construction and long-range dependencies.
- **INV3:** Measure robustness with made-up words in the sentences, ability to identify ARG0, ARG1 and ARG2 for both structure types.

Locative Alternation

Verbs part of this alternation patterns express placing substances on a surface or things into a container, or removing substances from a surface or from a container. There are two possible syntactic structures for this alternation: Locatum-as-Object or Location-as-Object sentence patterns. The former places the thing being moved or applied (locatum) in the direct object position, while the latter places the location in the direct object position. The variation in the sentence structure does not alter meaning but shifts the focus in the sentence (Levin, 1993). I used the *load* and *spray* verbs from VerbNet to handcraft the test instances for this capability, and PropBank was of help to ensure correct ARG1 and ARG2 label annotation.

Example

1. Locatum-as-Object: Alex smudged lipstick [ARG1] on his cheeks [ARG2].
2. Location-as-Object: Alex smudged his cheeks [ARG2] with lipstick [ARG1].

Test types and test description

- **MFT1:** Ability to identify ARG1 and ARG2 in Locatum-as-Object locative alternation pattern.
- **MFT2:** Ability to identify ARG1 and ARG2 in Location-as-Object locative alternation pattern.
- **INV1:** Ability to identify ARG1 and ARG2 in locative alternation patterns combined with long-range dependencies.

3.4 Subject Alternations

Instrument Subject Alternation

This is an alternation pattern where the instrument can take two different semantic roles. In this alternation pattern, the instrument can be either introduced using a prepositional phrase (*with a hammer*) and be a modifying adjunct, or it can take the agent role of the sentence. In the second case the emphasis shifts from

¹¹<https://verbs.colorado.edu/verb-index/vn/run-51.3.2.php>

¹²<https://verbs.colorado.edu/verb-index/vn/give-13.1.php>

person or thing doing the action to the instrument itself (Levin, 1993). The predicates in the created test instances belong to the verb class *break-45.1*¹³.

Example

1. He [ARG0] broke the window [ARG1] with a hammer [ARGM-MNR].
2. The hammer [ARG0] broke the window [ARG1].

Test types and test description

- **MFT1:** Ability to identify ARG0, ARG1 and ARGM-MNR in sentences where the instrument is introduced with a prepositional phrase.
- **MFT2:** Ability to identify ARG0 and ARG1 in sentences where the instrument is the agent.

3.5 Lexical variation

Metonymy

Metonymy could be defined as a figure of speech, in which one entity is used to refer to another entity, and the two are closely related to each other. For example, in the sentence *The trains are on strike.*, the subject of the sentence implies that it is train drivers who are striking (Littlemore, 2015). Metonymy relies on the common-sense knowledge and contextual interpretation of language users to draw these conceptual associations, which might be challenging to do for computational models. The MFT tests in this section aim to evaluate models' abilities to recognize when time expressions or locations serve as agents, and when they serve as modifiers of the predicate. DIR tests are applied to test the models' ability to discern between literal and metonymic use of expressions, and the resulting shift in argument labels.

Example

1. The world saw the beginning of a new era in 1989 [ARGM-TMP].
2. 1989 [ARG0] saw the beginning of a new era.

Test types and test description

- **MFT1:** Ability to identify locations in ARG0 role and time expressions in ARGM-TMP role.
- **MFT2:** Ability to identify time expressions in ARG0 role and locations in ARGM-LOC role.
- **DIR1:** Ability to recognize shift from literal to metonymic use of locations.
- **DIR2:** Ability to recognize shift from literal to metonymic use of time expressions.

¹³<https://verbs.colorado.edu/verb-index/vn/break-45.1.php>

4 Creating a Challenge Dataset

The complete test set consists of 490 test instances. There is a partial overlap within the test set, as some test types recycle sentences from others, either with a slight or no modification. Sentences for Locative Alternation, Preposition Drop Alternation and Metonymy were hand-crafted. ChatGPT-4¹⁴, a powerful large language model, was utilised to generate sentences for the remaining capabilities. When using ChatGPT-4, the prompt included a description of the capability being tested, the specific test type, an example sentence, and in some cases a list of verbs that should be used for test generation. An example of this prompt structure can be found in the Appendix. The generated output was then carefully reviewed, selected and annotated by me before being incorporated into the JSON file format.

Throughout the generation of test instances, I faced several challenges. Manually reviewing the large volume of output generated by ChatGPT-4 was time-consuming. Additionally, the online interface often had limitations, restricting output to a maximum of 10 sentences at a time before freezing. The generated output quickly became repetitive or ungrammatical, which was an issue to tackle in the process. While manual creation of test examples can result in higher quality of instances, it also places significant demands on the creativity of test designers. The manual approach resulted in a smaller number of examples for hand-crafted capabilities, potentially limiting the scope of conclusions that can be drawn from the results.

First, I inspected the models' recurring errors and evaluation scores per argument type on the original test set to identify areas where the model needs additional testing. Then, I consulted the work of a Levin (1993), who categorised verb alternations and verb classes. I leveraged the information available in PropBank and VerbNet, resources that provide information of argument structure per verb sense.

5 Evaluating Models

In this report three fine-tuned pre-trained DistilBERT-base-uncased models are evaluated using CHECKLIST. Sanh et al. (2019) introduce DistilBERT as the lighter and 40% smaller version of the BERT model (Devlin et al., 2018) with the intention to increase efficient execution of NLP tasks. Through knowledge distillation, DistilBERT maintains BERT's language processing capabilities while requiring less computational resources. In its name *base* refers to its size with six transformer layers, and *uncased* means that it does not

¹⁴provided by OpenAI <https://chat.openai.com>

Capability	Test Type and Description	Example test cases	#sent	Failure Rate (%)		
				M1	M2	M3
Active/passive voice	MFT (1): Identify ARG0 and ARG1 in short sentences with active voice	The cat [ARG0] <u>chased</u> the mouse [ARG1]. A child [ARG0] <u>found</u> a toy [ARG1].	20	0	0	0
	MFT (2): Identify ARG0 and ARG1 in short sentences with passive voice	The cake [ARG1] was eaten by the child [ARG0]. The book [ARG1] was <u>written</u> by the author [ARG0].	20	15	20	15
	MFT (3): Identify ARG0 and ARG1 active/passive sentences with 2 identical predicates	The chef [ARG0] <u>cooked</u> the meal [ARG1] and then cooked dessert to perfection.	20	0	0	0
	INV (1): Identify ARG0 and ARG1 passive sentences with long-range dependencies	The book [ARG1] was praised by critics [ARG0]. The book [ARG1], which was written over a decade ago, was praised by critics [ARG0].	10 pairs	0	0	0
	INV (2): Measure robustness with typos in the predicate, identify ARG0 and ARG1 in passive sentences	The case [ARG1] was <u>solved</u> by the detectives [ARG0]. The case [ARG1] was <u>sovled</u> by the detectives [ARG0].	10 pairs	0	0	0
	DIR (1): Identify shift from ARG0 to ARG2 due to preposition alternation in passive sentences	The car was <u>repaired</u> by the mechanic [ARG0]. The car was <u>repaired</u> for the mechanic [ARG2].	10 pairs	0	0	0
Causative-inchoative alt.	MFT (1): Short causative sentences with <i>break-45.1</i> , using volitional, non-volitional and instrument agents	John [ARG0] <u>broke</u> the window. The earthquake [ARG0] <u>shattered</u> the vase [ARG1].	20	10	10	10
	MFT (2): Short inchoative sentences with <i>break-45.1</i>	The window [ARG1] <u>broke</u> . The vase [ARG1] <u>fractured</u> .	20	0	0	0
	INV (1): Causative sentences with long-range dependencies with <i>break-45.1</i> , using volitional, non-volitional and instrument agents.	The child [ARG0] <u>shattered</u> the glass [ARG1]. The child [ARG0], acting with a mischievous intent that belied his innocent appearance, intentionally shattered the glass [ARG1] with a well-aimed throw.	10 pairs	0	0	0
	INV (2): Inchoative sentences with long-range dependencies with <i>break-45.1</i>	The car [ARG1] <u>crashed</u> . The car [ARG1], which I was driving down the hill, <u>crashed</u> .	10 pairs	0	0	0
	INV (3): Measure robustness with negation for inchoative sentences using <i>break-45.1</i>	The car [ARG1] <u>crashed</u> . The car [ARG1] <u>didn't crash</u> .	10 pairs	0	0	0
Preposition Drop alt.	MFT (1): Identify ARG0 and ARG1 in cases where ARG1 is specified with a PP, using <i>run-51.3.2</i>	Martha [ARG0] <u>climbed up</u> the mountain [ARG1].	10	30	50	60
	MFT (2): Identify ARG0 and ARG1 in cases where the preposition is "dropped", using <i>run-51.3.2</i>	Martha [ARG0] <u>climbed</u> the mountain [ARG1].	10	10	10	10
Dative alternation	MFT (1): Sentences with indirect object dative construction with <i>give-13.1</i>	He [ARG0] <u>gave</u> the book [ARG1] to John [ARG2]. She [ARG0] <u>let</u> the room [ARG1] to the visiting scholar [ARG2].	10	0	0	0
	MFT (2): Sentences with double object construction with <i>give-13.1</i>	He [ARG0] <u>gave</u> John [ARG2] the book [ARG1]. She [ARG0] <u>let</u> the visiting scholar [ARG2] the room [ARG1].	10	40	50	50
	INV (1): Sentences with indirect object dative construction and long-range dependencies, using <i>give-13.1</i>	He [ARG0] <u>gave</u> the keys [ARG1] to the owner [ARG2]. He [ARG0] <u>gave</u> the keys [ARG1], which had been found in the old desk drawer, to the owner [ARG2], who was eagerly waiting outside.	10 pairs	10	10	10
	INV (2): Sentences with double object construction and long-range dependencies, using <i>give-13.1</i>	The company [ARG0] <u>sold</u> the competitor [ARG2] the patent [ARG1]. The company [ARG0], which had been known for supporting innovation, <u>sold</u> the competitor [ARG2] the patent [ARG1], crucial for future innovations.	10 pairs	90	80	70
	INV (3): Measure robustness with made-up verbs with <i>give-13.1</i>	She [ARG0] <u>gave</u> the tennis ball [ARG1] to her friend [ARG2]. She [ARG0] <u>daxed</u> the tennis ball [ARG1] to her friend [ARG2].	10 pairs	20	20	20
Locative alt.	MFT (1): Identify ARG1 and ARG2 in Locatum-as-Object locative alternation pattern, using <i>spray/load</i> verbs	Josh <u>sprayed</u> paint [ARG1] on the wall [ARG2]. She <u>loaded</u> boxes [ARG2] into the wagon [ARG1].	20	65	70	60
	MFT (2): Identify ARG1 and ARG2 in Location-as-Object locative alternation pattern, using <i>spray/load</i> verbs	Josh <u>sprayed</u> the wall [ARG2] with paint [ARG1]. She <u>loaded</u> the wagon [ARG1] with boxes [ARG2].	20	90	90	90
	INV (1): Robustness: Identify ARG1 and ARG2, using <i>spray/load</i> verbs and long-range dependencies	The chef <u>sprinkled</u> the dish [ARG2] with salt [ARG1]. The chef <u>sprinkled</u> the dish [ARG2], which has just finished cooking, with salt [ARG1].	10 pairs	60	40	60
Instrument subject alt.	MFT (1): Identify ARG0, ARG1 and ARG2 where the instrument is introduced with a PP, using <i>break-45.1</i>	He [ARG0] <u>broke</u> the window [ARG1] with a hammer [ARG2].	20	0	0	5
	MFT (2): Identify ARG0 and ARG1 where the instrument is the agent, using <i>break-45.1</i>	The hammer [ARG0] <u>broke</u> the window [ARG1].	20	0	5	5
Metonymy variation	MFT (1): Ability to recognize locations acting as ARG0s and time expressions as ARGM-TMP	Washington [ARG0] <u>passed</u> the policy in 1978 [ARGM-TMP].	20	35	35	35
	MFT (2): Ability to recognize time expressions acting as ARG0s and locations as ARGM-LOC	January [ARG0] <u>brought</u> cold winds in Tokyo [ARGM-LOC].	20	90	75	75
	DIR (1): Ability differentiate between literal and metonymic uses of locations	She <u>won</u> the match in Manchester [ARGM-LOC]. Manchester [ARG0] <u>won</u> the match.	10 pairs	0	0	0
	DIR (2): Ability differentiate between literal and metonymic uses of time expressions	The world saw the beginning of a new era in 1989 [ARGM-TMP]. 1989 [ARG0] <u>saw</u> the beginning of a new era.	5 pairs	0	0	0

Table 2: A selection of CHECKLIST tests for Semantic Role Labeling (SRL)

distinguish between uppercase and lowercase tokens. It was trained in a self-supervised manner with the same training data (English Wikipedia and Toronto Book Corpus) as BERT, using the Masked Language Modeling objective. Despite its reduced size, DistilBERT nearly matches BERT’s performance, offering an accessible solution for various downstream tasks (Sanh et al., 2019).

In a previous project, three different DistilBERT models were fine-tuned¹⁵ for SRL using the Universal Proposition Banks version 1.0¹⁶ for English language. They share the commonality of predicting argument labels for each token in a given input sentence. The predicate token was marked in the input for each model, however, the method of marking slightly differs in complexity. Model 1 is considered to be the simplest baseline model, which will be compared to the other two advanced models using CHECKLIST.

Inspired by Shi and Lin (2019), the predicate token is presented for Model 1 by adding a special [SEP] token and the predicate word in the input sentence. In Model 2, we provide the context window of the predicate after the [SEP] token, which can help the model disambiguate which token is the predicate in case the sentence contains the same predicate token twice (Zhou and Xu, 2015). In Model 3, we directly mark the predicate token by inserting the [PRED] token immediately before the predicate (Khandelwal and Sawant, 2019).

Model Input Examples

- **Model 1:** Obama went to Paris last week. [SEP] went
- **Model 2:** Obama went to Paris last week. [SEP] Obama went to
- **Model 3:** Obama [PRED] went to Paris last week.

Model 1 expects a simple way of marking the predicate without information about the context and placement of the predicate token. For this reason, I expect that the baseline model will face challenges in sentences where the predicate token is present multiple times, or in cases where the predicate’s meaning changes depending on the context. Model 2 on the other hand, receives information about the words surrounding the predicate, which is likely to improve semantic role assignment in complex sentences and predicate disambiguation. Model 3 employs a strategy of clearly marking the predicate token for predicate

delineation and more accurate role prediction. Results using the original test set indicate that the advanced models perform slightly better compared to the baseline model, for more see Table 3 in the Appendix.

6 Results

Each model was tested individually, the results per test type are shown in Table 2. In this section I discuss the observed behavior of the models per capability. As opposed to the initial expectations, the advanced models reached similar failure rate scores compared to the baseline model.

Voice

The models had no difficulties identifying ARG0 and ARG1 in the designed active sentences, however, they did struggle with passive sentences that had increased syntactic complexity. For these erroneous passive sentences, the models did not manage to identify the agent when it was the last noun phrase in the sentence. An example where all models failed is: *The complex code [ARG1] was tirelessly debugged until it was flawless by the diligent programmer [ARG0]*.

Causative/Inchoative Alternation

The models tested well for this capability with *break-45.1* verbs, except for some sentences in the first MFT test, which contains causative sentences. The models failed to correctly label certain nouns that are relatively infrequent to have the ARG0 role in natural language, such as *strength* and *a power surge*.

Preposition Drop Alternation

Model 3 performed the weakest (60% failure rate) for the first MFT test with *run-51.3.2* verbs, followed by Model 2 (50% failure rate) and Model 1 (30% failure rate). The most common source of error was mistaking ARG1 for the locative modifier (ARGM-LOC), as in *The enthusiastic hikers [ARG0] walked along the trail [ARG1]*, or mistaking ARG1 for ARG2, e.g., *Martha [ARG0] climbed up the mountain [ARG1]*.

Dative Alternation

The models had difficulty identifying ARG2 correctly in double object constructions using *give-13.1* verbs within the second MFT test, they mainly mislabelled it with ARG1. For example, *The homeowner rented the vacationers [ARG2] the cottage [ARG1]*. The models had a failure rate above 70% for the second INV test, where sentences with double object constructions were enhanced using long-range dependencies. Here, most models not only mislabelled ARG2 for ARG1, but also missed to assign any label to ARG1

¹⁵batch size = 16, learning rate = 1e-4, epochs = 1

¹⁶<https://universalpropositions.github.io>

constituents at the end of the sentence. For example, *After weeks of negotiation, the bank [ARG0] finally loaned the small business owner [ARG2], eager to bring his vision to life, the money [ARG1]*. The models had 20% failure rate for the third robustness INV test, which targeted the models' capability to deal with made-up predicates.

Locative Alternation

Globally speaking, the models reached the highest failure rates, up to 90%, for this capability with *spray/load* verbs. In case of the first MFT test with Locatum-as-Object pattern, the models mislabelled ARG2 with ARGM-LOC, as in *The farmers scattered seeds [ARG1] across the field [ARG2]*. As for the second MFT test with Location-as-Object pattern, the models systematically mislabelled ARG1 with ARGM-MNR, for instance *The mother spreads the bagel [ARG2] with jam [ARG1]*. It also frequently occurred that the models confused ARG1 with ARG2 and vice versa, as in *The postman stuffed the envelopes [ARG2] with newsletters [ARG1]*.

Instrument Subject Alternation

The models performed well on the instrument alternation patterns challenged with *break-45.1* verbs except for a few mistakes. These resulted from mislabeling ARG0 with ARG1, in the instance *A whistle [ARG0] broke up the fight [ARG1]*.

Metonymy

Tests connected to this capability highlight that identifying locatives and time markers is challenging for the models. The first MFT test shows that the models had occasional errors when identifying time expressions in adjunct role. For example, the time marker was not labelled in *London [ARG0], despite facing an economic downturn, is continuing to invest heavily in public transportation in 2024 [ARGM-TMP]*.

The same applies for the second MFT test, with locations not being identified as adjuncts, e.g., *The 1920s [ARG0] roared with new fashion and trends in the USA [ARGM-LOC]*. The models mislabelled time expressions in ARG0 roles with ARGM-TMP labels at the beginning of sentences for some instances. An example illustrating this is the sentence: *Yesterday [ARG0] brought unexpected news to our doorstep in Istanbul [ARGM-LOC]*.

The DIR tests show no failed instances, since the models identified that the labels of location and time expressions changes depending on their position and role in the sentence. However, it is worth noting that the models missed to identify ARGM-LOC and

ARGM-TMP adjuncts in simple sentences, which means these scores should not be interpreted as indications of good performance.

7 Discussion

The CHECKLIST tests highlight that the models face limitations when handling certain syntactic and lexical capabilities, particularly with Dative Alternation, Locative Alternation, Preposition Drop Alternation, and labeling of ARGM-LOC and ARGM-TMP adjuncts. The deployment of MFT, INV and DIR test types provide an insight into understanding how the models tackle complex linguistic phenomena. Contrary to my expectations, the outcomes reveal that all three models perform similarly across the array of tests. Notably, the simplest model occasionally outperformed advanced models for some test types, such as the Preposition Drop Alternation MFT1.

The efficacy of the models in generalizing across diverse linguistic phenomena is strongly dependent on the training data provided in the PropBank v1.0 dataset. A possible explanation for the observed performance deficits is the lack of sufficient training instances for specific verb alternation patterns tested in this project. A further reason for such errors could be inconsistencies in the data annotation process. Certain argument labels are difficult to distinguish, such as ARG2 and ARGM-LOC, which can lead to human errors. The test instances in this dataset were created relying on the information available about verb argument structure PropBank, which might also contain discrepancies.

Robustness tests were implemented in this project, including addressing typos, long-range dependencies, negation and made-up predicates. The result of these tests reveal that the models exhibit a reasonable level of resilience against most forms of introduced noise. However, they show weaknesses when handling verb alternation patterns combined with complex syntactic structures characterized by multiple clauses. The models' performance could possibly be improved by additional model training targeting the identified weaknesses.

This project was executed under time constraints, which results in several restrictions. Expanding the range of capabilities, test types and test instances could provide a more nuanced understanding of the models' strengths and shortcomings. Aspects such as the models' interaction with idioms, dialects and language variation and fairness were beyond the scope of this study. The examination of most capabilities was confined to specific verb classes defined in VerbNet and utilized fewer than 20 sentences per test, limiting the ability to derive general conclusions. The test instances exclu-

sively focus on declarative sentences, experimenting with other sentence types could be part of future work.

8 Future Work

This section describes a scenario when the task is to perform SRL for the domain of doctor-patient conversations during a GP appointment, with limited amount of labelled data and a large corpus of unlabelled data available in the target domain. The task also specifies that we have access to abundant amount of data from additional domains.

Domain-specific Evaluation Set

The first step in this process is estimating the performance of a generic SRL system trained and tested on the universal proposition bank data on a domain-specific evaluation set. In order to accurately measure the performance of the SRL system in the target domain, the evaluation set should reflect the unique characteristics of doctor-patient conversations. This involves creating an evaluation set with annotated test examples representing domain-specific language, such as symptoms, diagnoses and patient concerns, thereby capturing the variety and specificity of the domain. Since the specified domain is characterised by spoken utterances transcribed into written form and medical vocabulary, the small amount of available labelled data could be utilised for creating the evaluation set. It is important that these examples cover a wide range of medical conditions, dialogue scenarios, and patient demographics to ensure the set reflects the real-world variety in doctor-patient conversations.

SRL System for the Target Domain

Depending on the outcomes of the evaluation, we might decide to refine the generic SRL system to reach better performance on the given domain. There are several domain adaptation strategies we can utilise for this test case. Assuming the pre-trained source model is available and we have a small amount of annotated in-domain data, **fine-tuning** offers a solution. During fine-tuning we transfer the pre-trained model to the target domain using continued model training, which adjusts the model's weights and parameters to bridge the gap between general language understanding and domain-specific knowledge (Laparra et al., 2020).

If the source model is not available, the first step could be training an SRL model using the large datasets available from other relevant domains, resulting in a model with broad understanding of language and semantic roles. Since we have a small amount of labelled data, we can utilise domain

adaptation techniques like **instance weighting and selection**. This technique involves selecting training instances that the model is most confident about and using these for further training, with the possibility of weighting them depending on their relevance to the target domain, and removing misleading source examples (Jiang and Zhai, 2007). The method called **neural feature augmentation** could also offer a solution. It enhances the feature space by adding domain-specific features from the labelled target dataset, such as instances with medical terminology or common semantic patterns in doctor-patient conversations. Lastly, **bootstrapping** is another strategy making use of the abundant in-domain unlabelled data. This means applying the pre-trained model to make predictions on unlabelled target data, selecting examples with a high confidence, and retraining the model with the extended set of examples. It is also possible to assign higher weight to the newly added target data (Wu et al., 2009). We could also consider implementing an interactive learning loop, where the model's predictions with high confidence score are reviewed by domain experts, and the corrections are fed back into the training process to ensure high-quality training data.

Iterative Evaluation

When implementing domain adaptation strategies, it is recommended to regularly evaluate the adapted SRL systems using a development set. This can help in identifying gaps in the model's understanding and decide which domain adaptation techniques lead to better outcomes. We can use the insights from these evaluation loops to refine the domain adaptation techniques and make more informed choices in the training process.

Ethical considerations

The handling of medical conversations introduces several ethical questions that must be considered to safeguard patient privacy and ensure fairness. Firstly, collecting patient data requires consent on behalf of patients and the medical staff. Given the nature of medical conversations, it is an essential step to mask or remove privacy-sensitive information, such as names, age, locations and personal identifiers. This process should follow the privacy standards and data anonymization guidelines, which are outlined in the Health Insurance Portability and Accountability Act (HIPAA) in the United States (Edemekong et al., 2018), or the General Data Protection Regulation (GDPR) in the European Union (Voigt and Von dem Bussche, 2017). We should also investigate demo-

graphic representation in the training data, which helps us establish if the models are likely to be biased, and whether they would amplify existing gender and racial disparities in healthcare. The evaluation set designed for model testing should include examples measuring both bias and fairness.

9 Conclusion

This project aimed to evaluate the syntactic and lexical competencies of three fine-tuned DistilBERT models for Semantic Role Labeling (SRL), employing the CHECKLIST methodology. Despite initial hypotheses suggesting that models with a more sophisticated representation of predicates might outperform the baseline model, findings indicate that performance does not significantly differ. For some test types, the simpler model demonstrated lower failure rates. The created instances cover a variety of test types including Minimum Functionality Tests (MFT), Invariance Tests (INV) and Directional Expectation Tests (DIR), each designed to test distinct model capabilities.

The failure rates across these diverse test types highlighted the consistent struggles of the models, particularly in accurately attributing semantic roles within sentences featuring Dative Alternation, Locative Alternation, Preposition Drop Alternation and metonymic shifts. The models notably stumbled when labelling sentences augmented with long-range dependencies, showing that handling complex sentence structures is a challenging area. Robustness tests, incorporating elements like negation and typographical errors alongside other linguistic capabilities, revealed a generic resilience against such perturbations.

The observable performance might stem from insufficient exposure to certain verb alternation patterns within the training dataset, or might be in connection with random weight initialization within the models. Continued training with a broader array of examples, particularly those covering the identified syntactic and lexical challenges, could improve the models' performance. Although this study provides an insight into the handling of complex and atypical sentence structures by the models, its scope is limited. We cannot draw general conclusions about the shortcomings of fine-tuned transformer-based models on SRL. Future research could explore alternative testing methodologies and models, including feature-based systems. The implemented CHECKLIST is only a starting point, which could be extended with more capabilities and test types covering a larger variety linguistic patterns. These tests could include exploring model interaction with idioms, dialects and a more thorough investigation of fairness.

Acknowledgements

Throughout the process of completing this assignment I collaborated with Murat Ertaş. This included sharing snippets of code, brainstorming about capabilities and design choices for the experiments, but did not include sharing any specific test types or test instances. I exchanged ideas about capabilities and literary resources with Christina Karavida.

References

- Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48.
- Xavier Carreras and Lluís Màrquez. 2005. [Introduction to the CoNLL-2005 shared task: Semantic role labeling](#). In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Peter F Edemekong, Pavan Annamaraju, and Micelle J Haydel. 2018. Health insurance portability and accountability act.
- Daniel Gildea and Daniel Jurafsky. 2002. [Automatic labeling of semantic roles](#). *Computational Linguistics*, 28(3):245–288.
- Sidney Greenbaum. 1996. *English grammar*. Oxford University.
- Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. *ACL*.
- Aditya Khandelwal and Suraj Sawant. 2019. Negbert: a transfer learning approach for negation detection and scope resolution. *arXiv preprint arXiv:1911.04211*.
- Egoitz Laparra, Steven Bethard, and Timothy A Miller. 2020. Rethinking domain adaptation for machine learning over clinical language. *JAMIA open*, 3(2):146–150.
- Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.
- Jeannette Littlemore. 2015. *Metonymy*. Cambridge University Press.
- Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2019. [Semeval-2016 task 4: Sentiment analysis in twitter](#).
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [Squad: 100,000+ questions for machine comprehension of text](#).
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Darsh J Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. [Adversarial domain adaptation for duplicate question detection](#).
- Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. [Predictive biases in natural language processing models: A conceptual framework and overview](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.
- Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed.*, Cham: Springer International Publishing, 10(3152676):10–5555.
- Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *EMNLP’09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 3*, pages 1523–1532. Association for Computing Machinery.
- Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural 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)*, pages 1127–1137.

Appendix

Date: 20/03/2024

Prompt:

Create 10 sentences just like in the example. The capability is Causative-inchoative alternation, the test type is MFT. Use the verbs from the verb class break-45.1: break, break apart, break down, break up, chip, cleave, crack, crash, crush, dissolve, fissure, fracture, fragment, rend, rip, shatter, shred, sliver, smash, snap, splinter, split, tear. Example: The chef deliberately cracked the egg.

Output:

1. The scientist carefully dissolved the compound.
2. The child accidentally shattered the vase.

3. The climber forcefully split the rock.
4. The gardener neatly snipped the branches.
5. The editor thoroughly shredded the documents.
6. The woodworker skillfully splintered the log.
7. The cook expertly snapped the asparagus.
8. The tailor precisely tore the fabric.
9. The jeweler meticulously slivered the metal.
10. The artist gently fractured the clay.

Sentences after review:

1. The scientist [ARG0] dissolved the compound [ARG1].
2. The child [ARG0] accidentally shattered the vase [ARG1].
3. The climber [ARG0] forcefully split the rock [ARG1].
4. The gardener [ARG0] neatly snipped the branches [ARG1].
5. The editor [ARG0] shredded the documents [ARG1].
6. The woodworker [ARG0] skillfully splintered the wood [ARG1].
7. The tailor [ARG0] tore the fabric [ARG1].
8. The girl [ARG0] fractured her bones [ARG1].

Aggregated Scores	Model 1	Model 2	Model 3
Accuracy	0.97	0.973	0.975
Macro Avg (F1)	0.404	0.42	0.417
Weighted Avg (F1)	0.969	0.972	0.974
Macro Avg (Recall)	0.399	0.416	0.414
Weighted Avg (Recall)	0.97	0.973	0.975
Macro Avg (Precision)	0.455	0.469	0.461
Weighted Avg (Precision)	0.969	0.972	0.974

Table 3: Performance scores using PropBank v1.0 test set