Systematicity, Compositionality and Transitivity of Deep NLP Models: a Metamorphic Testing Perspective

Anonymous ACL submission

Abstract

Metamorphic testing has recently been used to check the safety of neural NLP models. Its main advantage is that it does not rely on a ground truth to generate test cases. However, existing studies are mostly concerned with robustness-like metamorphic relations, limiting the scope of linguistic properties they can test. We propose three new classes of metamorphic relations, which address the properties of systematicity, compositionality and transitivity. Unlike robustness, our relations are defined over multiple source inputs, thus increasing the number of test cases that we can produce by a polynomial factor. With them, we test the internal consistency of state-of-the-art NLP models, and show that they do not always behave according to their expected linguistic properties. Lastly, we introduce a novel graphical notation that efficiently summarises the inner structure of metamorphic relations.

1 Introduction

Many recent advances in neural models for NLP have been driven by the ability to learn from unlabeled data (Devlin et al., 2019; Liu et al., 2019b). This approach allows for training the models on large-scale corpora without the costly process of annotating them. As a result, the accuracy and complexity of state-of-the-art neural models for NLP have increased (Brown et al., 2020).

This trend towards unlabeled data does not have a counterpart in testing NLP models. Instead, both in-distribution testing and out-of-distribution testing (Yin et al., 2019; Teney et al., 2020) rely on comparing the model's predictions to the ground truth. Similarly, attempts at *probing* the internal computation of large NLP models use supervised classifiers as a diagnostic tool (Ettinger et al., 2016; Belinkov et al., 2017).

In general, such extreme reliance on ground-truth data limits the quantity and quality of test cases we can produce, which is a known problem in the software testing community (Barr et al., 2015). In this regard, a promising solution is *metamorphic* testing (Chen et al., 2018). Under this paradigm, we test the internal consistency of an NLP model by checking whether it satisfies a necessary relation of its inputs and outputs (Ribeiro et al., 2020). Consequently, metamorphic testing relies on our ability to formally express our expectations on the behaviour of an NLP model.

Still, most of the metamorphic relations proposed in the literature target the same type of behaviour, as we show in this paper. Indeed, the majority of them are *robustness* relations, which require that the output of an NLP model remains stable in the face of small input perturbations (Aspillaga et al., 2020). These perturbations may involve simple typos (Belinkov and Bisk, 2018; Gao et al., 2018; Heigold et al., 2018), replacing individual words with a synonym (Li et al., 2017; Jia et al., 2019; La Malfa et al., 2020), or adding irrelevant information to the input (Tu et al., 2021). Due to their simple structure, robustness-like relations have been applied to the testing of several NLP tasks, including sentiment analysis (Ribeiro et al., 2020), machine translation (Sun and Zhou, 2018), and question answering (Chan et al., 2021). Even testing the fairness of NLP models falls in this category (Ma et al., 2020).

At the same time, we expect state-of-the-art NLP models to exhibit a broader range of linguistic properties than just robustness. First and foremost, NLP models should generalise *systematically*, i.e. their ability to understand some inputs should be intrinsically connected to their ability to understand related ones (Fodor and Pylyshyn, 1988). While the exact definition of systematic behaviour varies in the literature (Hupkes et al., 2020), a common requirement is that the model's predictions are a result of a *composition* of syntactic and semantic constituents of the input (Baroni, 2020). Several supervised methods to test against such requirements exist (Et-

tinger et al., 2016; Goodwin et al., 2020), but they all rely on comparing the model's predictions to the ground truth. Likewise, Yanaka et al. (2021) interprets systematicity as the ability to generalise over *transitive* relations. Their supervised method shows that current models struggle to do so.

In this paper, we propose three new classes of metamorphic relations, which are designed to test the systematicity, compositionality and transitivity of NLP models. In true metamorphic fashion, our relations do not rely on ground-truth data and scale up the generation of test cases by a polynomial factor. For each proposed relation, we provide an illustrative experiment where we test state-of-the-art models for the expected linguistic behaviours. More in detail, our main original contributions are:

- **Pairwise systematicity.** First, we propose a general class of metamorphic relations to test the systematicity of NLP models (Section 4). The relations in this class are based on *pairs* of inputs, which yields a quadratic number of test cases from a single dataset. We test the pairwise systematicity of a sentiment analysis model in Section 4.1, with positive results. Then, in Section 4.2, we give a geometrical intuition of the constraints imposed by our relations on the model's embedding space.

- **Pairwise compositionality.** Second, we modify pairwise systematicity to test the presence of compositional constituents in the hidden layers of neural models (Section 5). Accordingly, we test the pairwise compositionality of a natural language inference (NLI) model in Section 5.1, and show that it does not behave in a compositional way.

- **Three-way transitivity.** Third, we introduce a class of relations to test the internal transitivity of an NLP model (Section 6). These relations are defined over *triplets* of source inputs. In Section 6.1, we test a state-of-the-art model that predicts the lexical relation of words (synonymy, hypernymy), and show that it does not behave in a transitive way.

- **Graphical notation.** Fourth, we propose a formal graphical notation for NLP metamorphic relations, that efficiently expresses their internal structure (Section 2).

- **Taxonomy of existing work.** Fifth, we review the existing literature on metamorphic testing for NLP, and show that the relations proposed therein share the same structure with a single source input (Section 3).

Lastly, in Section 7 we conclude and outline possible future work. We discuss the ethical implications of our work in Appendix A. We provide a quick-reference guide to our contribution in Appendix B. The code of our experiments and reproducibility checklist are available at `https://doi.org/10.5281/zenodo.5703459`.

2 A graphical notation for NLP metamorphic relations

This section gives preliminary definitions and proposes a compact graphical notation for NLP metamorphic relations.

Definition 2.1 (NLP model)**.** Let $f : \mathcal{X} \to \mathcal{Y}$ be a machine learning model that maps a textual input $\mathbf{x} \in \mathcal{X}$ to a suitable output $\mathbf{Y} \in \mathcal{Y}$. Here, we assume that f is a neural network, and $\mathcal{Y} \equiv \mathbb{R}^k$ is either a k-dimensional embedding space or the soft-max output of a k-class classifier.

In general, a metamorphic relation can be defined as (Chen et al., 2018):

Definition 2.2 (Metamorphic relation)**.** A metamorphic relation R is a property of f across multiple inputs and outputs $(\mathbf{x}_1, \ldots, \mathbf{x}_v, f(\mathbf{x}_1), \ldots, f(\mathbf{x}_v))$, such that $R \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_v \times \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_v$.

However, we are interested in the internal structure of such a relation. Thus, let us discriminate between two types of inputs (Chen et al., 2018):

Definition 2.3 (Source inputs)**.** Given a relation R with v inputs, let $(\mathbf{x}_1, \ldots, \mathbf{x}_u)$ with $u \leq v$ be the sequence of source inputs. These can be chosen freely, e.g. by extracting them from a dataset $\mathcal{D}$.

Definition 2.4 (Follow-up inputs)**.** Given a relation R with u source inputs, let $(\mathbf{x}_{u+1}, \ldots, \mathbf{x}_v)$ with $u \leq v$ be the sequence of follow-up inputs. These are computed by a transformation of the source inputs $\mathbf{x}_i = T_i(\mathbf{x}_1, \ldots, \mathbf{x}_u)$ for $i \in [u+1, v]$.

Furthermore, all the relations in this paper prescribe specific conditions over the model's output:

Definition 2.5 (Output property)**.** Define $P \subseteq \mathcal{Y}_1, \ldots, \mathcal{Y}_v$ as a relation over the output. Here, we always write it in decidable first-order logic.

Altogether, the structure of an NLP metamorphic relation can be easily described in graphical

form. To do so, we introduce the following compact notation (see example in Figure 1). Textual variables are represented as circles, whereas numerical variables (e.g. embeddings, softmax outputs) are squares. Moreover, source inputs are shaded in grey, while all other nodes are in white. Arrows represent the neural function f and the transformation T_i. Lastly, the output property P is linked to the relevant nodes with dashed lines.

3 A taxonomy of existing NLP metamorphic relations

Most of the existing literature on NLP metamorphic testing proposes relations that fit in the structure of Figure 1. Due to their reliance on just one source input, we refer to these metamorphic relations as *single-input*. The individual differences among them can be ascribed to the specific transformation T and property P. The present section derives a taxonomy of existing NLP relations by organising them along these two axes T and P.

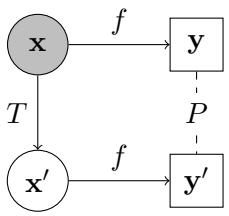

Figure 1: Structure of a single-input metamorphic relation. Property P expresses how the output of model f should change when the source input $\mathbf{x}$ is modified via T. Most relations in the literature follow this structure.

The transformation T is defined over the input text and thus allows for considerable creative freedom. A list of common options is presented here:

- **Character-level T.** Character-level transformations are typically used to introduce noise in the input. Examples include replacing individual characters with a neighbouring one on a computer keyboard (Belinkov and Bisk, 2018) or a random one (Heigold et al., 2018). More aggressive transformations may involve swapping neighbouring characters (Belinkov and Bisk, 2018; Gao et al., 2018; Heigold et al., 2018) and shuffling a subset of the characters in a word (Belinkov and Bisk, 2018). Alternatively, a collection of real-world typos can be retrieved from datasets with edit history (e.g. Wikipedia) (Belinkov and Bisk, 2018).

- **Word-level T.** A common word-level transformation involves replacing words with their synonym (Li et al., 2017). This operation has been shown to produce adversarial examples in (Jia et al., 2019; La Malfa et al., 2020). The use of antonyms has also been explored in Tu et al. (2021). In contrast, changing the gender of keywords in the input text can reveal the social biases of an NLP model (Ma et al., 2020). Similarly, swapping keywords in the context of a question-answer (QA) system can reveal inconsistent answers (Ribeiro et al., 2020).

- **Sentence-level T.** Removal or concatenation of entire sentences from the input text has been tried too. Aspillaga et al. (2020) experiments with adding positive and negative tautologies at the end of the input. Similarly, Ribeiro et al. (2020) propose to concatenate both well-formed sentences and randomly-generated URLs. More generally, the whole input text can have its sentences shuffled (Tu et al., 2021) or paraphrased (Li et al., 2017).

Regarding the output property P, the current literature only offers three choices. We list them here, alongside their first-order logic formulation:

- **Equivalence P.** Robustness relations require that the output does not change in the face of small input perturbations. Thus, we need a notion of equivalence between the source output $\mathbf{y}$ and its follow-up $\mathbf{y}'$ (see Figure 1). For classification models, we can express it via the softmax output $\mathbf{y} = (y_1, \dots, y_c)$ as:

$$P_{eq}: \quad \exists i \, \forall j \neq i \, (y_i > y_j) \wedge (y'_i > y'_j) \quad (1)$$

  where i is the predicted class. In rarer cases, where the output is textual, verbatim comparison can be used (Sun and Zhou, 2018).

- **Similarity P.** For other applications, the equivalence property cannot be applied. For example, when testing QA systems, we want to detect similar but not identical answers. In such cases, we can define a similarity score $s(\mathbf{y}, \mathbf{y}') \in \mathbb{R}$, e.g. cosine similarity between the embeddings of the two answers (Tu et al., 2021). With it, we can write similarity as:

$$P_{sim}: \quad s(\mathbf{y}, \mathbf{y}') > \theta \quad (2)$$

  where θ is an arbitrary threshold chosen according to the user's domain knowledge.

- **Order P.** At the same time, we can establish an order relation between the two outputs $\mathbf{y}$ and $\mathbf{y}'$. This order relation is useful in conjunction with transformations that have a monotonic effect on the output. For example, concatenating positive sentences to the input of a sentiment analysis system (Ribeiro et al., 2020). In such cases, let us define an order score $s(\mathbf{y}) \in \mathbb{R}$, and write the output property as:

$$P_{ord}: \quad s(\mathbf{y}) < s(\mathbf{y}') \tag{3}$$

In Sections 4, 5 and 6 we employ some of the transformations T and properties P defined here as building blocks for new metamorphic relations.

4 Pairwise NLP metamorphic relations for testing systematicity

We introduce a new class of metamorphic relations to test the systematicity of NLP models. Here, we take the general definition of systematicity in Fodor and Pylyshyn (1988), which states that the predictions of an NLP model across related inputs should be intrinsically connected and express it as a metamorphic relation (see Figure 2). Since we do not want to rely on ground-truth data, we first establish a baseline for the model's behaviour by comparing its predictions across two different source inputs. Then, we perturb both source inputs via the same transformation and test whether the model's behaviour changes accordingly.

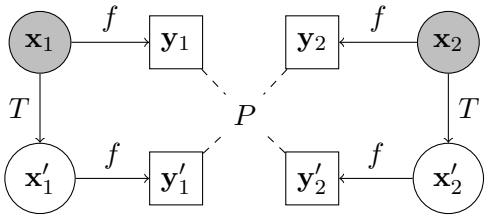

Figure 2: Structure of pairwise-systematicity relations. The two source inputs allow us to establish a baseline for the behaviour of model f, and test whether it changes according to expectations once T is applied.

More formally, we define *pairwise-systematicity* relations as follows. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$ be a pair of source inputs, and $\mathbf{x}'_1, \mathbf{x}'_2$ their corresponding follow-up inputs via transformation T. Furthermore, denote with $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}'_1, \mathbf{y}'_2$ the outputs produced by model f. Finally, define the output property P in the following form:

$$P: \quad P_{src}(\mathbf{y}_1, \mathbf{y}_2) \implies P_{flw}(\mathbf{y}'_1, \mathbf{y}'_2) \tag{4}$$

Note that this definition does not rely on ground-truth data. In fact, we trust the model's predictions $(\mathbf{y}_1, \mathbf{y}_2)$ over the source inputs to establish our premise P_{src}. The actual test checks whether transforming the source inputs with T produces outputs that satisfy the expected property P_{flw}. Any violation of this property, i.e. when $P_{src} \wedge \neg P_{flw}$, reveals an inconsistency in the model's predictions that breaks the user's expectation of systematic behaviour. In Section 4.2, we give an intuitive geometrical explanation of the type of constraints imposed by pairwise-systematicity relations on the embedding space of a neural NLP model.

A hidden advantage of metamorphic relations with multiple source inputs (see also Sections 5 and 6) is that they naturally produce more test cases than single-input ones. In the case of pairwise systematicity, each input in the pair $(\mathbf{x}_1, \mathbf{x}_2)$ is extracted from the same dataset $\mathcal{D}$. Thus, a dataset with $|\mathcal{D}| = k$ entries generates an $O(k^2)$ number of test cases, as opposed to $O(k)$ for single-input relations. We see an example of this in Section 4.1.

4.1 Illustrative example: pairwise systematicity of sentiment analysis

Now, let us apply the pairwise-systematicity relation structure shown in Figure 2 to a sentiment analysis task. To do so, we choose the following:

- **Transformation T.** For each source input $\mathbf{x}_i$, we create a follow-up input $\mathbf{x}'_i = T(\mathbf{x}_i)$ by concatenating a short sentence to it. A list of all transformations we use is in Table 1.

- **Output premise P_{src}.** Let $s_{pos}(\mathbf{y}_1)$ and $s_{pos}(\mathbf{y}_2)$ be the (positive) sentiment scores predicted by model f. Define the baseline behaviour of f as the order property $P_{src} = P_{ord}$ between these two scores (see Equation 3).

- **Output hypothesis P_{flw}.** Let $s_{pos}(\mathbf{y}'_1)$ and $s_{pos}(\mathbf{y}'_2)$ be the sentiment scores of the follow-up inputs. We require that their order matches the one of the source inputs. More formally: $P_{flw} = P_{ord}$ and $P_{src} \iff P_{flw}$.

Our rationale is that the sentiment of any input shifts when we concatenate additional text. If we have ground-truth information on the sentiment of the text we are adding, we can test whether our predictions shift in the expected direction. For instance, concatenating *"I am very happy"* should make the score of any input more positive. This is

Safety	Concatenated Text	Position
0.900	My friends were happy, though.	End
0.910	Anyway, the sound of the rain outside was soothing.	End
0.922	As always: popcorn and coke make everything better!	End
0.932	Thank you.	Start
0.943	I watched this movie with my brother.	Start
0.955	Here is my review:	Start

Table 1: Input transformations sorted by safety.

Safety	Source Input	Pred.
0.731	This isn't a "Friday" worth waiting for.	Pos
0.741	The audience when I saw this one was chuckling at all the wrong times, and that's a bad sign when they're supposed to be having a collective heart attack.	Pos
...	...	...
1.000	As a director, Paxton is surprisingly brilliant, deftly sewing together what could have been a confusing and horrifying vision into an intense and engrossing headtrip.	Neg
1.000	Intended to be a comedy about relationships, this wretched work falls flat in just about every conceivable area.	Pos

Table 2: Source inputs and their sentiment predictions, sorted by the number of times they appear in a safe pair.

an example of single-input relation (see Section 3 and Ribeiro et al., 2020).

However, if we do not have such ground truth, we can still test our model. We do so by considering a pair of inputs $(\mathbf{x}_1, \mathbf{x}_2)$, and concatenating the same text to both of them. Then, whenever $\mathbf{x}_1$ is predicted more positive than $\mathbf{x}_2$, we require that its transformed version $\mathbf{x}'_1$ is also more positive than $\mathbf{x}'_2$ and vice versa. This is pairwise systematicity.

Experiment description and results. We select a fine-tuned version of RoBERTa (Liu et al., 2019b) for sentiment analysis from the HuggingFace library.[1]. We choose 10,605 movie reviews from Socher et al. (2013) as our dataset $\mathcal{D}$. From it, we generate all 112M+ possible source input pairs. We repeat our experiment with different neutral transformations T, and report their aggregated results in Table 1. Note how the proportion of satisfied relations ("Safety") varies across different transformations. Yet, the model's behaviour is fairly systematic, never exceeding 10% violations.

We get a different picture by counting the number of violations per each source input $x_i \in \mathcal{D}$ (see Table 2). There, we can see that some inputs are more likely to make the source order $P_{src}(\mathbf{y}_1, \mathbf{y}_2)$ unstable across all the transformations T. Interestingly, a quick read through the reviews in Table 2 shows that they are all misclassified. Thus, we can conclude that pairwise-systematicity testing reveals a different issue in the model f than classic non-metamorphic testing. For this reason, we encourage practitioners to perform both types of testing on their NLP models, as it will give a clearer picture of their strengths and weaknesses.

[1] https://huggingface.co/siebert/sentiment-roberta-large-english

4.2 Geometric interpretation of pairwise systematicity

Metamorphic relations impose constraints between the inputs and outputs while treating the model f as a black box (Chen et al., 2018). Still, in neural networks, it is possible to trace the effect of a relation R on the hidden layers. Here, we give a geometric explanation of the type of constraints pairwise-systematicity relations put on the last embedding space of a neural NLP model.

To this end, let us consider the relations in Section 4.1. Recall, that model f outputs a sentiment score $s(\mathbf{y})$, which is a one-dimensional projection of the embedding space (see Figure 3). Accordingly, the premise P_{src} and hypothesis P_{flw} are only concerned with the position of each embedding $\mathbf{y}$ along direction s. However, since the source and follow-up inputs differ due to transformation T, the two output properties P_{src} and P_{flw} act on different points in the embedding space. Once we require that $P_{src} \iff P_{flw}$, we set the expectation that f is exceptionally consistent at mapping pairs of inputs $(\mathbf{x}_1, \mathbf{x}_2)$ onto space $\mathcal{Y}$ in the same order.

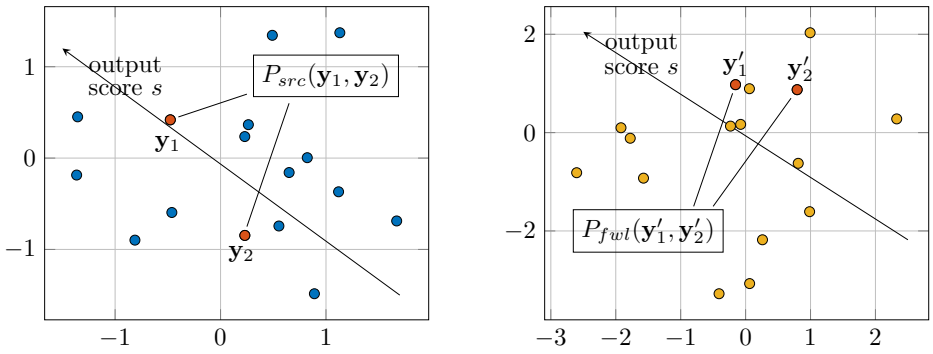

Figure 3: Pairwise systematicity relates pairs of source outputs (left) to pairs of follow-up outputs (right) in the embedding space. For the relations in Section 4.1, the order of each pair along dimension s must be preserved.

Such expectation is met if and only if f is a systematic, though not necessarily correct, function.

Similar considerations apply if P_{src} and P_{flw} are based on equality or similarity rather than order. Indeed, equality (see Equation 1) is defined over the softmax outputs, which are affine combinations of the embeddings (Bishop, 2006). In such case, the condition $P_{src} \implies P_{flw}$ translates to a requirement that if the source inputs are both mapped to the same half-space, the follow-up inputs should be too. Conversely, similarity (Equation 2) defines a measure on the embedding space. Source inputs that are within a certain threshold θ should be matched by follow-up inputs that are also close.

The following section introduces a class of pairwise relations where the output premise and hypothesis are defined over different embedding spaces.

5 Pairwise NLP metamorphic relations for testing compositionality

Many *probing* works train simple supervised classifiers on top of the hidden representations of an NLP model (e.g. Hewitt and Manning, 2019).These classifiers, called *probes*, can reveal whether the neural model has learnt to recognise some fundamental constituents of the input language early on. The presence of such building blocks is a necessary condition for an NLP model to exhibit compositional behaviour (Baroni, 2020). Here, we propose to test the presence of compositional constituents in the hidden layers via metamorphic testing.

Consider the graph in Figure 4. There, the neural model is split into the mathematical composition of two functions $f \circ g$. More precisely, $\mathbf{z} = f(\mathbf{x})$ are the hidden representation of some hidden layer, and $\mathbf{y} = g(\mathbf{z})$ is the final output. Now, let us define the output property P as follows:

$$P: \quad P_{hid}(\mathbf{z}_1, \mathbf{z}_2) \implies P_{out}(\mathbf{y}_1, \mathbf{y}_2) \quad (5)$$

A relation in this form allows us to express whether specific precursor signals in $\mathbf{z}$ are expected to have a direct effect on $\mathbf{y}$. In a similar way to the relations in Section 4, both the premise P_{hid} and hypothesis P_{out} are established by comparing across pairs of inputs, rather than a ground-truth. In Section 5.1, we show how our technique can reveal the presence (or absence) of compositional building blocks in an NLP model.

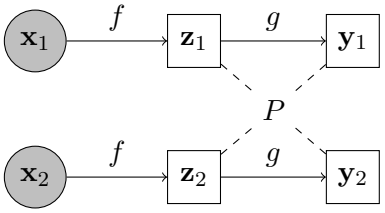

Figure 4: Structure of pairwise-compositionality relations. Comparing the hidden representations $\mathbf{z}_1, \mathbf{z}_2$ of the source inputs reveals whether the model $f \circ g$ uses them to produce the output in a compositional fashion.

5.1 Illustrative example: pairwise compositionality of NLI

Here, we apply the metamorphic relation in Figure 4 to test a natural language inference (NLI) model. In general, the input $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$ of an NLI model is the concatenation of two pieces of text: the premise $\mathbf{x}_a$ and the hypothesis $\mathbf{x}_b$. The model's goal is to predict whether $\mathbf{x}_b$ logically follows from $\mathbf{x}_a$, i.e. their *entailment*.

To test whether the model's predictions exhibit a compositional behaviour, we construct our test inputs according to Rozanova et al. (2021). Namely,

Safety	Context	Mon.
0.387	So there is no dedicated $\langle x \rangle$ for every entity and no distinction between entity mentions and non-mention words.	Down
...	...	...
0.626	There was no $\langle x \rangle$.	Down
0.627	We stood on the brink of a $\langle x \rangle$.	Up
...	...	...
0.746	There are some old houses in this $\langle x \rangle$.	Up
0.754	Some $\langle x \rangle$ bloom in spring and others in autumn.	Up

Table 3: Contexts sorted by increasing safety.

Safety	Insertion Pair	Lex. Rel.
0.417	(gun,woman)	none
0.475	(woman,gun)	none
0.508	(tree,cherry tree)	$\supseteq$
...	...	...
0.590	(fruit,apple)	$\supseteq$
0.591	(pine,tree)	$\subseteq$
...	...	...
0.696	(potatoes,animals)	none
0.726	(animals,potatoes)	none

Table 4: Insertions sorted by increasing safety.

we first choose a prototypical sentence template $C(\ell)$, which we call a *context*. Each context includes a placeholder token ℓ that can be replaced with some *insertion* text. Second, we construct each input $\mathbf{x} = (C(\ell_a), C(\ell_b))$ by copying the same context twice with different insertions.

Finally, we choose the contexts C_i and insertion pairs $(\ell_a, \ell_b)_j$ in such a way that their composition $(C(\ell_a), C(\ell_b))_{ij}$ has a well-definite entailment relation. Namely, the insertion pairs (see Table 4) are either hypernyms ($\supseteq$), hyponyms ($\subseteq$), or unrelated (none). Similarly, the contexts (see Table 3) are either *upward monotone* if they preserve the insertion relation, or *downward monotone* if they invert it. As a result, only the compositions Up($\subseteq$) and Down($\supseteq$) are entailed, while the rest are not.

Now, assume that both inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ in Figure 4 are based on the same context C_i. We can test whether the NLI model build its output by reasoning over the monotonicity of C_i and the lexical relation of the insertion pairs $(\ell_a, \ell_b)_j$ as follows:

- **Hidden premise P_{hid}.** Let $\mathbf{z}$ be the embeddings of the second to last layer, for the tokens corresponding to the insertions ℓ_a and ℓ_b. Train a linear probe s_{hyp} on $\mathbf{z}$ (Liu et al., 2019a) to predict whether ℓ_a is a hypernym of ℓ_b. Define $P_{hid} = P_{ord}$ as the order property (see Equation 3) over the hypernymy scores $s_{hyp}(\mathbf{z}_1)$ and $s_{hyp}(\mathbf{z}_2)$ of the two inputs.

- **Output hypothesis P_{out}.** Let $s_{ent}(\mathbf{y})$ be the entailment score produced by the full neural model $f \circ g$. Moreover, define $P_{out} = P_{ord}$ as the order of the two output scores $s_{ent}(\mathbf{y}_1)$ and $s_{ent}(\mathbf{y}_2)$. Then, consider the monotonicity of the input context. If C_i is downward monotone, let $P_{hid} \iff P_{out}$, since more hypernymy means more entailment. If C_i is upward monotone, let $P_{hid} \iff \neg P_{out}$, since more hypernymy means less entailment.

If the NLI model $f \circ g$ had a compositional behaviour, the order P_{hid} of the hypernymy scores in the hidden layer should be reflected in the order P_{out} of the entailment scores in the output. Here, we show that this is not the case for a popular state-of-the-art NLI model.

Experiment description and results. We build a dataset $\mathcal{D}$ of 292 insertions pairs and repeat our experiment with 211 contexts, for a total of about 9M test cases. We chose a fine-tuned version of RoBERTa for NLI as our model.[2] The accuracy of the hypernymy probe is 0.9881. We report the aggregated result by context in Table 3. Note how downward monotone contexts lead to less compositional behaviour: overall, we have 0.6880 successful test cases with upward contexts and only 0.4808 with downward ones. This phenomenon is known in the literature (Yanaka et al., 2019), but we show that metamorphic testing can independently detect it. If we aggregate the results by insertion pair (see Table 4), the picture does not change. The overall safety is 0.5936, which is barely above random chance. Any deviations from this baseline can be interpreted as noise.

6 Three-way NLP metamorphic relations for testing transitivity

An NLP model that generalises correctly should exhibit *transitive* behaviour under the right circumstances Yanaka et al. (2021). That is, if the model

[2] https://huggingface.co/roberta-large-mnli

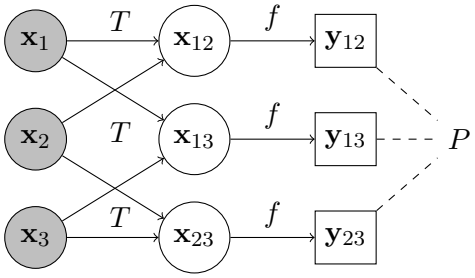

Figure 5: Structure of three-way transitivity relations. The three source inputs $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are combined into all possible pairs. If two pairs are predicted as true by model f, the third must be predicted true as well.

predicts a transitive linguistic property over the input pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_2, \mathbf{x}_3)$, then it should also predict it for the pair $(\mathbf{x}_1, \mathbf{x}_3)$. Here, we propose to test this behaviour in a metamorphic way.

More specifically, let us introduce the *three-way transitivity* relation in Figure 5. There, the three source inputs $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are combined to form all possible input pairs $\mathbf{x}_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$. Then, we can test whether their corresponding outputs are transitive with the following output property:

$$P: \quad v(y_{12}) \wedge v(y_{23}) \Rightarrow v(y_{13}) \qquad (6)$$

where $v(\cdot) : \mathcal{Y} \rightarrow \{0, 1\}$ is the Boolean prediction of model f. Note that the output property P, being defined over three outputs, has a different structure from those in Sections 3, 4 and 5.

6.1 Illustrative example: three-way transitivity of lexical relations

In this section, we apply the metamorphic structure from Figure 5 to test the transitivity of *lexical semantic relations*, e.g. synonymy and hypernymy (Santus et al., 2016). In general, learning these linguistic properties is crucial for solving several NLI tasks (Glockner et al., 2018). Thus, we can expect an NLP model to generalise over them in a transitive way. We can test whether this is true in the following way:

- **Transformation T.** The model f we test already accepts a pair of words $\mathbf{x}_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$ as input. Thus, T is merely a formalism here.

- **Output Property P.** Property P in Equation 6 depends on the definition of $v(\cdot)$. Here, we train two classification heads on top of a pretrained model f. The first $v_{syn}(\cdot)$ predicts synonymy, the second $v_{hyp}(\cdot)$ hypernymy.

Language	Syn. Safety	Hyp. Safety
English	0.191	0.277
German	0.240	0.287
Chinese	0.390	0.394
Italian	0.341	0.259

Table 5: Proportion of successful three-way transitivity tests for a state-of-the-art lexical relation model.

Note that transitivity can be tested in a supervised fashion by comparing the model's predictions to a ground truth (Yanaka et al., 2021). In contrast, the three-way transitivity relations we propose test the *internal* transitivity of a model trained to predict lexical relations.

Experiment description and results. We reproduce a state-of-the-art model for lexical relations (Wachowiak et al., 2020), which is a fine-tuned version of the multi-lingual transformer model `xlmroberta` (Conneau et al., 2020). We extract the multi-lingual test set from the CoGALex_VI shared task (Santus et al., 2016), and generate a random sample of source triplets from its corpus of words, keeping those that satisfy $v(y_{12}) \wedge v(y_{23})$. We present our empirical results in Table 5, organised by the language of the source words and lexical relation v predicted by the model. As the table shows, this state-of-the-art NLP model fails to predict $v(y_{13})$ in a transitive way across all languages. This is in contrast with the results of classic supervised testing in Wachowiak et al. (2020), which show that their model can predict the correct lexical relations (synonym, hypernym, antonym or random) with at least 0.5 of accuracy.

7 Conclusions and future work

In this paper, we presented three new classes on metamorphic relations. Thanks to them, we could test the systematicity, compositionality and transitivity of state-of-the-art NLP models. The advantage of our approach is that it does not rely on ground-truth annotations. It can generate a polynomially larger number of test cases than supervised testing, revealing whether the NLP model under test is internally consistent.

Still, testing is only one side of the coin. Like in recent work about robustness (Aspillaga et al., 2020), the tested models have not been trained on a metamorphic objective (e.g. as an additional loss term). We believe that doing so could improve the safety and consistency of a model's predictions.

References

Carlos Aspillaga, Andrés Carvallo, and Vladimir Araujo. 2020. Stress test evaluation of transformer-based models in natural language understanding tasks. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1882–1894, Marseille, France. European Language Resources Association.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. 2015. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41(5):507–525.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Alvin Chan, Lei Ma, Felix Juefei-Xu, Yew-Soon Ong, Xiaofei Xie, Minhui Xue, and Yang Liu. 2021. Breaking neural reasoning architectures with metamorphic relation-based adversarial examples. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–7.

Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. 2018. Metamorphic testing: A review of challenges and opportunities. *ACM Comput. Surv.*, 51(1).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online. Association for Computational Linguistics.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 68–80, Boston, MA. Association for Machine Translation in the Americas.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Emanuele La Malfa, Min Wu, Luca Laurenti, Benjie Wang, Anthony Hartshorn, and Marta Kwiatkowska. 2020. Assessing robustness of text classification through maximal safe radius computation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2949–2968, Online. Association for Computational Linguistics.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27, Valencia, Spain. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Pingchuan Ma, Shuai Wang, and Jin Liu. 2020. Metamorphic testing and certified mitigation of fairness violations in NLP models. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 458–465. ijcai.org.

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, Online. Association for Computational Linguistics.

Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Supporting context monotonicity abstractions in neural NLI models. *CoRR*, abs/2105.08008.

Enrico Santus, Anna Gladkova, Stefan Evert, and Alessandro Lenci. 2016. The CogALex-V shared task on the corpus-based identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 69–79, Osaka, Japan. The COLING 2016 Organizing Committee.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria. Association for Computational Linguistics.

Liqun Sun and Zhi Quan Zhou. 2018. Metamorphic testing for machine translations: Mt4mt. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 96–100.

Damien Teney, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton van den Hengel. 2020. On the value of out-of-distribution testing: An example of goodhart's law. In *Advances in Neural Information Processing Systems*, volume 33, pages 407–417. Curran Associates, Inc.

Kaiyi Tu, Mingyue Jiang, and Zuohua Ding. 2021. A metamorphic testing approach for assessing question answering systems. *Mathematics*, 9(7).

Lennart Wachowiak, Christian Lang, Barbara Heinisch, and Dagmar Gromann. 2020. CogALex-VI shared task: Transrelation - a robust multilingual language model for multilingual relation identification. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 59–64, Online. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. Exploring transitivity in neural NLI models through veridicality. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 920–934, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach.

In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Appendix A. Ethics statement

Intelligent systems are becoming increasingly widespread, and NLP models are often used as important components in their architecture. However, once these systems are deployed in the real world, there is a risk of them exhibiting biased, erratic or dangerous behaviour. In order to prevent such events from happening, it is crucial to perform a thorough testing and validation process. Indeed, this is one of the tenets of the ACM Code of Ethics and Professional Conduct[3]. Namely, paragraph 2.5 therein recites *"Extraordinary care should be taken to identify and mitigate potential risks in machine learning systems."* The contributions we propose in the present paper are directed towards this goal. More specifically, we believe that metamorphic testing is a valuable tool in the model tester's arsenal, and our contributions widen its scope of application. As a result, more instances of unwanted behaviour can be identified and addressed before their impact is felt by the end user.

Appendix B. Quick-reference guide

In this paper, we discuss and compare four classes of metamorphic relations. For ease of reference, we summarise them in Tables 6, 7, 8 and 9. These tables contain the formal definitions of the transformation T and output property P, a concrete example of possible inputs, and a reference to the corresponding sections in the present paper.

[3] https://www.acm.org/code-of-ethics

Single-input metamorphic relations	
Input:	$\mathbf{x}$ = The cat sat on the mat.
	$\mathbf{x}'$ = The pet stood onto the mat.
T:	*replace any word of the input with a synonym.*
P:	$\mathbf{y} = f(\mathbf{x}) \wedge \exists i \forall j \neq i \, (y_i > y_j) \wedge (y'_i > y'_j)$

Table 6: Example of robustness relations from the literature (Li et al., 2017). Robustness relations belong to the class of single-input relations (see Section 3).

Pairwise systematicity metamorphic relations	
Input:	$\mathbf{x}_1$ = Light, cute and forgettable.
	$\mathbf{x}_2$ = A masterpiece four years in the making.
	$\mathbf{x}'_1$ = Thank you. Light, cute and forgettable.
	$\mathbf{x}'_2$ = Thank you. A masterpiece four years in the making.
T:	*concatenate the text* Thank you. *at the beginning of the input.*
P:	$s_{pos}\big(f(\mathbf{x}_1)\big) > s_{pos}\big(f(\mathbf{x}_2)\big) \iff s_{pos}\big(f(\mathbf{x}'_1)\big) > s_{pos}\big(f(\mathbf{x}'_2)\big)$

Table 7: Example of pairwise systematicity relations defined on a sentiment analysis task (see Section 4.1).

Pairwise compositionality metamorphic relations	
Input:	$\mathbf{x}_1$ = There was no tree. There was no cherry tree.
	$\mathbf{x}_2$ = There was no fruit. There was no apple.
Hidden:	$f(\mathbf{x}_1)$ = *contextual embeddings of the tokens (* tree. cherry tree. *)*
	$f(\mathbf{x}_2)$ = *contextual embeddings of the tokens (* fruit. apple. *)*
P:	$s_{hyp}\big(f(\mathbf{x}_1)\big) > s_{hyp}\big(f(\mathbf{x}_2)\big) \iff s_{ent}\big(g(f(\mathbf{x}_1))\big) > s_{ent}\big(g(f(\mathbf{x}_2))\big)$

Table 8: Example of pairwise compositionality relations defined on a natural language inference task (see Section 5.1). Pairwise compositionality relations do not have a transformation T.

Three-way transitivity metamorphic relations	
Input:	$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ = arrangement symmetrical together
	$\mathbf{x}_{12}$ = *(* arrangement symmetrical *)*
	$\mathbf{x}_{23}$ = *(* symmetrical together *)*
	$\mathbf{x}_{13}$ = *(* arrangement together *)*
T:	*choose two words from the source triplet* $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$
P_{syn}:	$v_{syn}\big(f(\mathbf{x}_{12})\big) \wedge v_{syn}\big(f(\mathbf{x}_{23})\big) \implies v_{syn}\big(f(\mathbf{x}_{13})\big)$
P_{hyp}:	$v_{hyp}\big(f(\mathbf{x}_{12})\big) \wedge v_{hyp}\big(f(\mathbf{x}_{23})\big) \implies v_{hyp}\big(f(\mathbf{x}_{13})\big)$

Table 9: Example of three-way transitivity relations defined on the lexical relations of synonymy and hypernymy (see Section 6.1).