It Is Not Easy To Detect Paraphrases: Analysing Semantic Similarity With Antonyms and Negation Using the New *SemAntoNeg* Benchmark

Teemu Vahtola and **Mathias Creutz** and **Jörg Tiedemann**
Department of Digital Humanities
Faculty of Arts
University of Helsinki
Finland
{teemu.vahtola, mathias.creutz, jorg.tiedemann}helsinki.fi

Abstract

We investigate to what extent a hundred publicly available, popular neural language models capture meaning systematically. Sentence embeddings obtained from pretrained or fine-tuned language models can be used to perform particular tasks, such as paraphrase detection, semantic textual similarity assessment or natural language inference. Common to all of these tasks is that paraphrastic sentences, that is, sentences that carry (nearly) the same meaning, should have (nearly) the same embeddings regardless of surface form.

We demonstrate that performance varies greatly across different language models when a specific type of meaning-preserving transformation is applied: two sentences should be identified as paraphrastic if one of them contains a negated antonym in relation to the other one, such as *I am not guilty* versus *I am innocent.*

We introduce and release *SemAntoNeg*, a new test suite containing 3152 entries for probing paraphrasticity in sentences incorporating negation and antonyms. Among other things, we show that language models fine-tuned for natural language inference outperform other types of models, especially the ones fine-tuned to produce general-purpose sentence embeddings, on the test suite. Furthermore, we show that most models designed explicitly for paraphrasing are rather mediocre in our task.

1 Introduction

Large pretrained language language models have pushed NLP forward in many sub-fields, and their ability to embed essential linguistic properties makes them applicable across a wide range of tasks. However, it is still an open question how well they cope with systematic compositionality and to what level of abstraction they reflect the actual meaning behind a given sentence.

This work is in line with other experiments based on dedicated test suites that study specific linguistic phenomena in connection with neural representation models. In particular, we publish a novel benchmark called *SemAntoNeg*[1], that tests the ability of language models to properly represent expressions that contain antonymy and negation, embedded into a paraphrase task.

Our test suite consists of contrastive examples where the task is to select the correct paraphrase for a given input sentence among three candidate expressions that include combinations of negated sentences and antonym substitutions. To provide a simple example, a semantic opposite to an input sentence *I am guilty* would be *I am not guilty*, where the opposite meaning is invoked by an insertion of a negation marker. Similarly, instead of inserting the negation marker, substituting the adjective to its antonym inverses the meaning of the sentence: *I am innocent*. To maintain paraphrasticity with respect to the original sentence, performing both of the operations is necessary, resulting in: *I am not innocent*. Thus, for a model to succeed in the *SemAntoNeg* test suite, the models need to understand that insertion or deletion of a negation accompanied with antonym substitution produces a sentence that is semantically equivalent to the original sentence, and the sentence embeddings should represent this relationship.

Using this benchmark, we study the following questions:

- How well do sentence embeddings from general-purpose language models fare in this task?
- Does fine-tuning on paraphrase tasks help to improve the performance on our test suite?
- What is the best fine-tuning task that supports our benchmark?

[1]The challenge set is available at: https://github.com/teemuvh/antonym-substitution

Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 249 - 262

In order to answer those questions, we systematically evaluate a large number of publicly available pretrained language models on the novel test suite. We notice a large amount of variation between different models, mostly depending on the fine-tuning objective and data, but, in some cases, also between models fine-tuned for the same task. Selecting an appropriate model becomes a challenge considering the sea of releases available on public model hubs. Simply selecting among the most popular ones might be a poor strategy, as we can see in Table 1.

Model	Accuracy
all-MiniLM-L6-v2	1.9
paraphrase-MiniLM-L6-v2	43.3
bert-base-nli-mean-tokens	83.3
all-mpnet-base-v2	31.9
distiluse-base-multilingual-cased-v2	1.5
all-MiniLM-L12-v2	8.2
multi-qa-mpnet-base-dot-v1	17.5
paraphrase-multilingual-MiniLM-L12-v2	61.8
paraphrase-mpnet-base-v2	76.1
distilbert-multilingual-nli-stsb-quora-ranking	47.1

Table 1: Testing our new benchmark on the ten most downloaded models from the Hugging Face sentence-transformers library in a descending order based on download counts.

Our analysis below provides new insights into the semantic abstraction abilities with respect to antonyms and negation and gives additional guidance for the selection of an appropriate model for tasks that require proper inference with such constructions.

2 Description of the Task

The objective of our test suite is to test to what extent sentence representation models succeed in distinguishing sentences similar in meaning when the change in semantics is realised only by a substitution of a distributionally similar (Grefenstette, 1992) word token (antonym in this case) or by an insertion or deletion of a negation marker. Performing either of the operations results in a sentence that conveys the semantic opposite of the original sentence while maintaining a high lexical overlap between both sentences.

We cast the benchmark in terms of a paraphrase detection task: a model is confronted with three alternatives of potential reformulations of an input sentence and only one of them is a proper equivalent on a semantic level. The candidates are designed to include negations and antonyms of adjectives to create the specific challenge of the test suite. More details about the data sets and its creation are given in Section 3 below.

The idea is not to fine-tune any model for this particular task (because any model would quickly overfit to such regular constructions) but rather to test independently trained sentence embedding models that can be used to measure semantic distance to make the decision. As such, it is in line with other natural language understanding benchmarks such as SentEval (Conneau and Kiela, 2018) and GLUE (Wang et al., 2018) but it represents a dedicated linguistic probing task rather than a general-purpose evaluation framework.

The work is motivated by previous research that identified deficiencies in the popular sentence representation benchmarks. For instance, existing paraphrase detection data sets (e.g., QQP[2]) lack examples that are characterised by a high lexical overlap without paraphrastic meaning (Zhang et al., 2019). Classification models can simply learn to measure lexical overlap to make proper decisions and, therefore, Zhang et al. (2019) generate a more difficult test suite of paraphrases and non-paraphrases with a high bag-of-words overlap by word scrambling. The same problem has been observed in natural language inference, where contradicting sentences typically exhibit low lexical overlap. Word permutations have been proposed (Dasgupta et al., 2018) to generate difficult cases that require better knowledge of compositionality.

In our test suite, we go beyond the creation of more challenging distractors (e.g., better negative examples with high vocabulary overlaps) by introducing more challenging positive candidates that are explicitly different from the source by adding antonyms from a lexical resource. In connection with negation, the model is now forced to disregard surface features (such as matching tokens) and to properly understand negated messages to make the correct decision. Thus, we test not only for semantic similarity but also evaluate the ability to understand the relationship between antonyms as well as the effect of negation on meaning representations.

Before discussing the experimental setup (Section 4) and the results of our practical experiments (Section 5), we will present details of the data set and how it was created in the section below.

[2] https://www.kaggle.com/c/quora-question-pairs

Label	Input	Options
2	No, that's true.	{No, that's false., No, that's not true., No, that's not false.}
2	I'm guilty.	{I'm innocent., I'm not guilty., I'm not innocent.}
2	I'm not sure.	{I'm not uncertain., I'm sure., I'm uncertain.}
2	That is good.	{That is bad., That is not good., That is not bad.}
2	I know you're not asleep.	{I know you're not awake., I know you're asleep., I know you're awake.}

Table 2: Examples from the test suite. Based on the input sentence, the model is supposed to select the equivalent sentence from the alternative hypotheses in the Options column. Values on the Label column indicate the index of the true paraphrase in the options column.

3 Data Creation

We created the test suite in a semi-automatic way. First, we downloaded 1.5 million sentence pairs from the English training set of the Opusparcus paraphrase corpus (Creutz, 2018). Next, we removed sentence pairs where the length of either of the sentences was less than four tokens. Furthermore, we retained only sentence pairs, where either one, but not both, of the sentences contains a negation (e.g., *I'm innocent. – I'm not guilty.*) After this filtering process, our data consisted of approximately 7500 sentence pairs. At this point we realised that, even though many of the sentence pairs were meaningful for our experiments, our filtered set also contained pairs that were paraphrastic but did not include antonymous relations, such as *Aren't you cold? – Are you cold?* Therefore, we proceeded with a second round of filtering, where we POS-tagged the sentences using NLTK (Bird et al., 2009) and retained sentences containing explicit negation (*not*) as an adverb (RB) as well as adjectives (JJ). Finally, we removed duplicate sentences. This process resulted in 1317 sentences, all of which included an explicit negation marker and an adjective.

For each of the 1317 sentences, we generated three hypotheses from which the model is supposed to choose the one that conveys the same meaning as the input sentence. First, we queried an antonym for the adjective in the input sentence from the WordNet Electrical Lexical Database (Fellbaum, 1998) to get a an opposing sentence. To obtain the second contradicting sentence, we deleted the negator from the input sentence. Finally, we substituted the adjective from the second contradicting sentence to its antonym to obtain the paraphrase of the input sentence. Examples of entries in the test suite are provided in Table 2.

We reviewed the resulting test suite manually to ensure its good quality. The antonym substitution procedure introduced some grammatical errors to the data, such as wrong agreement of articles (*a evil idea*) or a question tag not agreeing with the main clause (*You're not serious, aren't you?*). We corrected such phrases manually. The sentences also included some examples where the automatically retrieved antonyms were not considered to carry opposite meanings, such as *Are you hungry? – Are you not thirsty?* We removed such examples from the final test suite.

Eventually, we obtained 788 examples, from which we permuted all possible input sentences to result in 3152 test examples, containing 209 unique adjectives, which constitute the *SemAntoNeg* test suite.

4 Experimental Setup

To compare sentence representations derived from different Transformer-based pretrained language models, we ran 114 of the 120 pretrained Sentence-BERT (Reimers and Gurevych, 2019) models that are publicly available in the Hugging Face transformers library (Wolf et al., 2020).[3] We provide the full list of the models we tested accompanied with the accuracy they acquired on the *SemAntoNeg* test suite in Appendix A.

We embed the sentences in our test suite using the different language models. To create the embeddings we apply the same pooling strategy used in training the original sentence-transformers. We then evaluate each model by its ability to produce embeddings such that the input sentence and its true paraphrase are closest to each other in the vector space. To compare embeddings we use cosine similarity.

Basically, we have run a systematic loop over all

[3]We did not include four image-to-text models, nor did we include two T5-xxl models that we could not fit into the GPU.

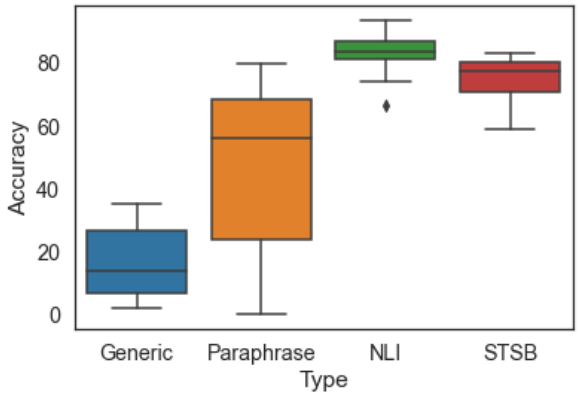

Figure 1: Results of the different model types. The accuracy [%] of the models is labeled on the y-axis and the different model types are labeled on the x-axis. "Generic" refers to the general-purpose models fine-tuned for a large range of transfer tasks. "Paraphrase", "NLI", and "STSB" refer to models fine-tuned for paraphrase detection, natural language inference, and semantic textual similarity, respectively.

available models but we are especially interested in specific groups of models, which we will discuss next. First of all, we want to see the performance of general-purpose models before looking at dedicated paraphrase models. Finally, we discuss other task-specific models before we present a detailed error analysis and conclude with prospects for future work.

5 Results

We analyse the results of our experiments by grouping the language models into four different categories, determined by the fine-tuning objective of the models: general-purpose embeddings, as well as embeddings specialised for paraphrasing, natural language inference and semantic textual similarity. Figure 1 shows average accuracies and variances for these four types of models. Individual results of all models are provided in Appendix A.

Since there are three possible choices in our task, a random selection yields a baseline level of 33.3% accuracy. We observe models that diverge clearly from the baseline level, either positively or negatively. This means that many tested models make good or bad choices systematically. We return to these findings in the error analysis in Section 6. The following sections summarise the analyses of the separate model groups.

Model	Accuracy
sentence-t5-xl	81.2
sentence-t5-large	78.4
sentence-t5-base	67.2
all-roberta-large-v1	35.4
all-mpnet-base-v2	31.9
all-mpnet-base-v1	25.1
all-distilroberta-v1	18.5
all-MiniLM-L12-v1	9.7
all-MiniLM-L12-v2	8.2
all-MiniLM-L6-v1	3.1
all-MiniLM-L6-v2	1.9

Table 3: Results of the general-purpose models. The dashed line represents results from random choice (33.3%)

5.1 General-Purpose Sentence Representations

Table 3 provides results on the general-purpose sentence representation models. These models are trained to generate representations that have the capacity to be highly useful for a large range of natural language understanding tasks. When extensively evaluated on different benchmarks related to sentence embeddings and semantic search, the aggregated results reported on the sentence-transformers website indicate good performance.[4] The performance on the *SemAntoNeg* test suite probing for understanding of negation and antonymy, however, suggests that especially the fine-tuned general-purpose models lack in this crucial aspect of natural language understanding.

The T5 models (Raffel et al., 2020) are not fine-tuned for any specific objective. The all-* models, instead, are fine-tuned for a wide range of natural language understanding tasks and are thus expected to produce general-purpose sentence embeddings that have the capacity to capture a diverse range of linguistic properties in order to be successful in different transfer tasks. Analysis of the predicted sentences from the only fine-tuned general-purpose model that outperforms the random baseline (all-roberta-large-v1) suggests that the model is prone to predicting the sentence with the highest lexical overlap while ignoring the negation. Thus, the model often predicts the semantic opposite of the input sentence by simply connecting lexical similarity with semantic similarity, which explains the

[4] https://www.sbert.net/docs/pretrained_models.html

Model	Accuracy
paraphrase-multilingual-mpnet-base-v2	79.9
paraphrase-mpnet-base-v2	76.1
paraphrase-TinyBERT-L6-v2	72.1
paraphrase-distilroberta-base-v2	68.4
quora-distilbert-base	67.2
paraphrase-multilingual-MiniLM-L12-v2	61.8
paraphrase-albert-small-v2	60.8
paraphrase-albert-base-v2	58.7
paraphrase-MiniLM-L12-v2	55.9
quora-distilbert-multilingual	47.1
paraphrase-MiniLM-L6-v2	43.3
paraphrase-xlm-r-multilingual-v1	23.9
xlm-r-distilroberta-base-paraphrase-v1	23.9
paraphrase-distilroberta-base-v1	19.4
paraphrase-MiniLM-L3-v2	0.03

Table 4: Results [%] of the models fine-tuned for paraphrase detection. The dashed line represents results from random choice (33.3%)

Model	Accuracy
roberta-large-nli-mean-tokens	93.6
roberta-base-nli-mean-tokens	89.5
bert-large-nli-mean-tokens	87.4
nli-bert-large-cls-pooling	86.8
nli-bert-large-max-pooling	86.4
xlm-r-large-en-ko-nli-ststb	85.4
xlm-r-bert-base-nli-mean-tokens	83.9
bert-base-nli-cls-token	83.7
bert-base-nli-mean-tokens	83.3
nli-distilbert-base-max-pooling	83.2
distilbert-base-nli-mean-tokens	81.2
bert-base-nli-max-tokens	80.8
nli-roberta-base-v2	79.6
nli-mpnet-base-v2	74.1
nli-distilroberta-base-v2	66.6

Table 5: Results [%] of the models that are fine-tuned for natural language inference.

poor performance.

5.2 Paraphrase Models

We report the results of the paraphrase models on the test suite in Table 4. One would expect the paraphrase models, whose objective is to recognise sentences that carry the same meaning but use a different wording (Bhagat and Hovy, 2013) to be successful on our test suite. This is, however, not the case for all of the models, and none of the models actually outperform the best performing general-purpose model.

The paraphrase models seem to vary greatly in their ability to generate representations that capture meaning regardless of surface form. Some of the differences between the results can be traced back to the data used for fine-tuning the models. For instance, the two distilled implementations of RoBERTa (Liu et al., 2019) perform very differently on the test suite (19.4% vs. 68.4%). However, "version 2" (paraphrase-distilroberta-base-v2), which obtains the higher accuracy, was fine-tuned with much more training examples[5].

Some of the variation can be traced back to the methodology, such as with the MiniLM models (Wang et al., 2020). The difference in performance between the MiniLM models may be explained by the layer from which the representations are extracted. Previous research has demonstrated how Transformer-based models accumulate different types of knowledge on different layers, semantics being predominantly encoded in the last layers of the network (Jawahar et al., 2019). The results obtained for the MiniLM models are in line with the previous findings: embeddings from layer 12 outperform embeddings from layer 6, which in turn outperform embeddings from layer 3. To a smaller extent, the same effect is seen on the general-purpose models (Table 3) where the representations derived from the MiniLM models perform differently based on the layer they are extracted from.

5.3 Other Fine-Tuning Objectives

In addition to paraphrasing, we hypothesise that models which have been fine-tuned on other similar objectives could perform well on the *SemAntoNeg* test suite.

Table 5 provides results for all models fine-tuned for natural language inference (NLI) that are available in the sentence-transformers library. Natural language inference probes the model for recognising whether an input sentence (the premise) entails, contradicts or is neutral with respect to another sentence (the hypothesis).

Compared to the general-purpose models (Table 3) and the paraphrase models (Table 4), the NLI models are, for the most part, considerably more successful on the test suite. The success of the NLI models might arise due to a more prominent use of negation in the training data, giving a model more knowledge about the proper treatment of such constructions. Furthermore, models trained for NLI have been shown to understand the effect of ex-

[5] https://www.sbert.net/examples/training/paraphrases/README.html

Model	Accuracy
stsb-bert-large	83.3
stsb-roberta-large	82.4
stsb-roberta-base	80.3
stsb-xlm-r-multilingual	79.8
stsb-bert-base	77.3
stsb-distilbert-base	76.5
stsb-roberta-base-v2	70.8
stsb-mpnet-base-v2	70.2
stsb-distilroberta-base-v2	58.7

Table 6: Results [%] of the models fine-tuned for the Semantic textual similarity benchmark.

Model	Accuracy
roberta-large-nli-mean-tokens	93.6
roberta-base-nli-mean-tokens	89.5
bert-large-nli-mean-tokens	87.4
nli-bert-large-cls-pooling	86.8
nli-bert-large-max-pooling	86.4
xlm-r-large-en-ko-nli-ststb	85.4
xlm-r-bert-base-nli-mean-tokens	83.9
bert-base-nli-cls-token	83.7
bert-base-nli-mean-tokens	83.3
stsb-bert-large	83.3
bert-large-nli-stsb-mean-tokens	83.3
distilbert-base-nli-max-tokens	83.2

Table 7: Results [%] of the best performing models.

plicit negation (i.e., *not*) to the sentence semantics rather well (Kim et al., 2019). Additionally, entailment examples may get very close to instances in our test suite and, in this way, provide better support for the expressions we test. Paraphrase data sets, on the other hand, tend to emphasise the use of synonyms and therefore fail to learn a better treatment of negation and antonyms. Even though NLI models seem to be successful on our test suite, they do not necessarily perform well on other paraphrase tasks. BERT-large fine-tuned for NLI obtains 87.4% accuracy on the *SemAntoNeg* test suite, while only reaching 75.9% on the Microsoft Research Paraphrase Corpus (Reimers and Gurevych, 2019), where the best-performing models fine-tuned explicitly for paraphrasing obtain results exceeding 90% accuracy (e.g., fine-tuned RoBERTa achieves 92.3% on that task (Liu et al., 2019)). In future work, we will investigate the qualitative difference in training data in more depth in order to provide a better picture about the influence of fine-tuning objectives on *SemAntoNeg* performance.

In addition to NLI, we believe that fine-tuning sentence representation models on the Semantic Textual Similarity Benchmark (STSB) data (Cer et al., 2017) can produce embeddings that can distinguish between semantically similar sentences regardless of their surface forms and be successful on the test suite. The results of the models fine-tuned for STSB are presented in Table 6. The STSB data is designed to comprise sentences that share some level of semantic similarity, and the task probes the representations for "gradations of meaning overlap" (Agirre et al., 2016; Cer et al., 2017). Fine-tuning on the STSB data potentially encourages models to learn more fine-grained (dis-)similarities from the sentence pairs, which is valuable for succeeding in the *SemAntoNeg* test suite.

The sentence-transformers library also includes models that are not suited to the *SemAntoNeg* task by design. Such models include for instance models trained for machine reading comprehension and question answering on the MS MARCO data set (Bajaj et al., 2016). The results of these models are included in Appendix A, and affirm the hypothesis that the models are not suitable for this task.

6 Error Analysis

We have analysed misclassified examples from a set of different models. Naturally, the errors the best-performing models make differ from the ones made by the worst-performing models.

We have studied an intersection of the examples that were misclassified by all of the best models (listed in Table 7). Antonym pairs that are rare and highly contextual seem to be difficult for the models. For instance, the antonym pair *possible – actual* (e.g., in the sentence pair *No, that's actual. – No, that's not possible.*) is very often misclassified. The antonym pair *possible – actual* comprises the majority of the common misclassified examples of the best performing models: 70 out of 105 examples. The antonym pair is retrieved from WordNet, but in the test suite they rarely occur in a natural context (in which they would refer to a possibility, as opposed to an actual event). The example would probably demand some contextual priming for the models to be able to connect the relationship between the antonyms.

Another frequently shared incorrectly predicted antonym pair includes the words *same* and *other*

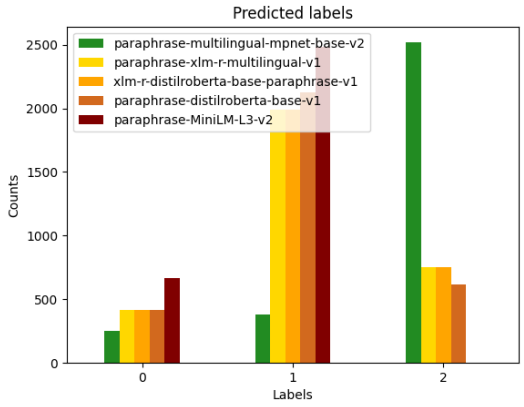

Figure 2: The best and the four worst performing paraphrase models. Labels in the test suite are presented on the x-axis. The number of times the labels are predicted by the models is presented on the y-axis. Labels 0 and 1 denote errors, 0 meaning sentences where only antonym substitution is performed, and 1 meaning only polarity swap (insertion or deletion of negation). Label 2 indicates that both operations are performed, which is the correct choice.

in the definite form, for instance: *It is not the same. – It is the other.* Here, the definite form *the other* could be the cause of incorrect predictions, whereas a more suitable opposing relation could be expressed using for example *some* as an indefinite article as in *some other*. The antonym pair *the same – the other* comprises 17 of the remaining 35 incorrectly predicted examples shared by the best performing models.

A bar chart of the best paraphrase model and the four paraphrase models that obtain accuracy below random choice is presented in Figure 2. The results indicate that the poor performance of the worst paraphrase models is explained by them systematically preferring the sentence with the highest lexical overlap disregarding the negation completely, which is reflected by a high proportion of label 1 in the figure. The trend seems similar for the other models whose accuracies are below random choice. Compared to the models that perform well, the poorly performing models seem more prone to associating lexical similarity with semantic similarity, leading them to predict the sentence with the opposite meaning with respect to the input sentence.

7 Limitations

The test suite comes with some limitations that we find important to discuss. We acknowledge that the proposed task is not difficult to solve using a simple rule-based model and that it is rather easy to overfit a neural network-based model to the data. Therefore, the kind of data the test suite instantiates is supposed to be used for evaluating models by probing sentence representations for the certain kinds of linguistic phenomena exclusively.

The test suite includes overrepresentation of certain frequent adjectives. The adjective *good* occurs more than 270 times in the input sentence, whereas some rarer adjectives, such as *opaque* only occur twice. Adding more unique adjectives to make the data even more representative and balancing the data is left for future work.

Another caveat of the test suite is that for now it only probes the sentence representations for a set of negated antonyms that belong to the adjective class. As some of the other word classes also include words that have a related opposing concept (e.g., *accept – reject* in the verb class), and it would be equally important to assess how language models understand the relation of the words in other word classes. Additionally, the test suite consists only of one certain negation pattern: not + adjective. Adding examples with more variable negation patterns would require an adapted filtering method to extract the sentences from a paraphrase corpus or more manual work to ensure high-quality paraphrases of the sort (e.g., *I walked – I didn't stand still.*) Augmenting the test suite with test examples of more varied negation patterns, as well as antonymous tokens from different word classes is left for future work.

8 Related Work

Previous work has focused on understanding negation, on the one hand, and antonymy, on the other.

Kassner and Schütze (2020) show that pretrained language models (Transformer-XL (Dai et al., 2019), ELMO (Peters et al., 2018) and BERT (Devlin et al., 2019) in this case) are poor at recognising the difference between a sentence in affirmative or negative form when they are queried with a negative cloze test, and are prone to predict the same token regardless of the polarity of the sentence. Ettinger (2020) study how BERT understands negation with similar minimal pairs to our test suite. However, they probe BERT for predicting one word token in a sentence pair where one sentence is a negated version of the other (e.g., *Most smokers find that quitting is very __. – Most smokers find that quitting isn't very __.*) BERT does not seem to

be highly robust for the transformation, and it does not seem that negation alone suffices to prime the model for systematically predicting the opposite of the prediction in the affirmative sentence (Ettinger, 2020).

Hossain et al. (2020) show that NLI models are not robust to negation by analysing models on a new benchmark designed to assess how models understand negation. In a similar spirit, assessing a multilingual language model fine-tuned for NLI on a test suite of minimal pairs of label-changing and label-preserving negations, Hartmann et al. (2021) find that multilingual language models are not fully aware of the effects that a negation marker can have on sentence semantics. Furthermore, NLI models' difficulty to represent negations reliably has been traced to training data, suggesting that models trained on SNLI (Bowman et al., 2015) or MNLI (Williams et al., 2018) do not properly learn to reason with expressions that include negation (Geiger et al., 2020; Richardson et al., 2020).

Kim et al. (2019) analyse different pretraining objectives for predicting textual entailment on various function word probing tasks, one of which assesses models' understanding of negation in a similar manner to our test suite. They find that the natural language inference models outperform other pretraining objectives in representing negation, mostly owing to NLI models' capability to represent explicit negation. However, analysis of examples that were difficult for a state-of-the-art NLI model has suggested that antonymy and negation are challenging phenomena to represent reliably, as models do not recognise antonymous relations as semantically opposing and may associate explicit negation to contradiction in neutral or entailed examples (Naik et al., 2018).

In addition to analysing language models' understanding of natural language in textual entailment, representation of antonyms has been studied for instance by comparing the mapping of negated adjectives in vector space (Rimell et al., 2017). BERT has also been adapted to perform a cloze problem for predicting antonyms in context (Niwa et al., 2021).

Additionally, previous research has focused on learning vector-based representations of word semantics that can model the relationship between distributionally similar but semantically opposing words better (e.g., Pham et al., 2015; Ono et al., 2015). Jumelet and Hupkes (2018) study how language models understand semantic compositionality with respect to contrasting meanings but focus on transformations of (negative) polarity items.

9 Conclusions

We have presented a novel test suite, *SemAntoNeg*, designed to probe pretrained language models for the understanding of the relationship between negation and antonymy. Contradicting examples in the test suite are close to the input by design and lead to a challenging benchmark. In order to succeed on our test suite, a model needs to recognise the semantic opposites invoked either by antonym substitution or by an insertion or a deletion of a negation marker. Equally, the model needs understanding of semantic compositionality to understand how the operations affect semantics of the sentence when performed together.

We have evaluated publicly available pretrained sentence representation models and reported results that display a large amount of variation when assessed on the new test suite. Surprisingly, dedicated paraphrase models are not among the best performing models and deliver rather poor results in many cases, whereas fine-tuning to natural language inference seems very beneficial for the task. General-purpose models are overall not very good at recognising our examples either, except for recent very large multi-task models such as T5-xl.

Our findings highlight that models that fare well in established natural language understanding benchmarks may still have crucial deficiencies in representing certain, rather typical, linguistic constructions and may produce critical mistakes. As a result, more structured test suites are necessary for assessing how the pretrained models understand language. This paper provides another contribution in that direction.

There are various avenues in future work we would like to explore. First of all, we need to further test the scaling effects when moving to very large language models such as GPT-3 (Brown et al., 2020). The T5 results already indicate that size matters but it is too early to draw general conclusions. Furthermore, we plan to investigate prompting as an alternative to vector similarity. However, prompt engineering is a challenging task in itself and we will need to explore the influence of prompts on results we can expect. Finally, we would also like to move to other languages and potentially cross-lingual setups.

Acknowledgements

The research reported in this paper was supported by the Behind the Words project, funded by the Academy of Finland. We would like to acknowledge *CSC – The Finnish IT Center for Science* for the computational resources they have generously provided. We would also like to thank the anonymous reviewers for their insightful comments.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Rahul Bhagat and Eduard Hovy. 2013. What Is a Paraphrase? *Computational Linguistics*, 39(3):463–472.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. *CoRR*, abs/1802.04302.

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. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Gregory Grefenstette. 1992. Finding semantic similarity in raw text: The deese antonyms. In *Fall Symposium Series, Working Notes, Probabilistic Approaches to Natural Language*, pages 61–65.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *Proceedings of the 25th Conference on*

Computational Natural Language Learning, pages 244–257, Online. Association for Computational Linguistics.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ayana Niwa, Keisuke Nishiguchi, and Naoaki Okazaki. 2021. Predicting antonyms in context using BERT. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 48–54, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989, Denver, Colorado. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8713–8721.

Laura Rimell, Amandla Mabona, Luana Bulat, and Douwe Kiela. 2017. Learning to negate adjectives with bilinear models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 71–78, Valencia, Spain. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Results of All Models

Results of the 114 Transformer-based models from the Hugging Face Transformers library on the test suite.

Model	Accuracy
msmarco-distilbert-base-tas-b	0.0
multi-qa-MiniLM-L6-cos-v1	0.3
all-MiniLM-L6-v2	1.9
multi-qa-distilbert-cos-v1	1.4
all-MiniLM-L12-v2	8.2
all-distilroberta-v1	18.5
multi-qa-mpnet-base-dot-v1	17.5
all-mpnet-base-v2	31.9
paraphrase-MiniLM-L3-v2	0.0
paraphrase-albert-small-v2	60.8
sentence-t5-base	67.2
distiluse-base-multilingual-cased	1.5
distilroberta-base-msmarco-v1	1.7
nli-bert-large-cls-pooling	86.8
xlm-r-base-en-ko-nli-ststb	79.5
bert-large-nli-cls-token	86.8
nli-distilbert-base-max-pooling	83.2
nli-bert-large-max-pooling	86.4
xlm-r-bert-base-nli-mean-tokens	83.9
msmarco-roberta-base-v2	4.3
distilbert-base-nli-max-tokens	83.2
xlm-r-100langs-bert-base-nli-mean-tokens	83.9
msmarco-MiniLM-L-12-v3	0.0
msmarco-MiniLM-L12-cos-v5	0.0
nli-distilbert-base	81.2
xlm-r-large-en-ko-nli-ststb	85.4
quora-distilbert-base	67.2
facebook-dpr-question_encoder-single-nq-base	6.8
facebook-dpr-question_encoder-multiset-base	5.9
nli-bert-base	83.3
bert-large-nli-max-tokens	86.4
msmarco-roberta-base-ance-firstp	3.1
bert-base-nli-cls-token	83.7
stsb-bert-large	83.3
facebook-dpr-ctx_encoder-multiset-base	9.4
bert-large-nli-stsb-mean-tokens	83.3
multi-qa-MiniLM-L6-dot-v1	0.5
msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.0
nli-roberta-base-v2	79.6
nli-roberta-base	89.5
stsb-distilroberta-base-v2	58.7
bert-base-wikipedia-sections-mean-tokens	7.1
stsb-bert-base	77.3
paraphrase-albert-base-v2	58.7
msmarco-distilbert-base-dot-prod-v3	0.4
msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.9

bert-large-nli-mean-tokens	87.4
xlm-r-distilroberta-base-paraphrase-v1	23.9
msmarco-roberta-base-v3	3.1
bert-base-nli-max-tokens	80.8
distilbert-base-nli-stsb-quora-ranking	67.2
msmarco-MiniLM-L6-cos-v5	0.0
msmarco-distilroberta-base-v2	0.4
nli-distilroberta-base-v2	66.6
roberta-base-nli-mean-tokens	89.5
distilroberta-base-paraphrase-v1	19.4
msmarco-MiniLM-L-6-v3	0.0
distilroberta-base-msmarco-v2	0.4
nq-distilbert-base-v1	2.3
msmarco-distilbert-cos-v5	0.0
msmarco-distilbert-base-v2	0.7
msmarco-distilbert-base-v3	0.0
stsb-xlm-r-multilingual	79.8
allenai-specter	0.1
roberta-large-nli-stsb-mean-tokens	82.4
roberta-base-nli-stsb-mean-tokens	80.3
use-cmlm-multilingual	3.6
xlm-r-100langs-bert-base-nli-stsb-mean-tokens	79.8
stsb-roberta-base	80.3
msmarco-bert-base-dot-v5	0.0
quora-distilbert-multilingual	47.1
stsb-roberta-large	82.4
xlm-r-bert-base-nli-stsb-mean-tokens	79.8
paraphrase-MiniLM-L12-v2	55.9
clip-ViT-B-32-multilingual-v1	Image-text model
msmarco-distilbert-dot-v5	0.0
nli-mpnet-base-v2	74.1
paraphrase-TinyBERT-L6-v2	72.1
distiluse-base-multilingual-cased-v1	1.4
distilbert-base-nli-stsb-mean-tokens	76.5
stsb-roberta-base-v2	70.8
paraphrase-distilroberta-base-v1	19.4
bert-base-nli-stsb-mean-tokens	77.3
LaBSE	11.7
stsb-distilbert-base	76.5
paraphrase-distilroberta-base-v2	68.4
paraphrase-multilingual-mpnet-base-v2	79.9
distilbert-base-nli-mean-tokens	81.2
distilbert-multilingual-nli-stsb-quora-ranking	47.1
msmarco-distilbert-base-v4	0.0
paraphrase-xlm-r-multilingual-v1	23.9
distiluse-base-multilingual-cased-v2	1.5
paraphrase-mpnet-base-v2	76.1
paraphrase-multilingual-MiniLM-L12-v2	61.8
paraphrase-MiniLM-L6-v2	43.3
bert-base-nli-mean-tokens	83.3
clip-ViT-B-16	Image-text model

clip-ViT-L-14	Image-text model
clip-ViT-B-32	Image-text model
sentence-t5-xxl	–
sentence-t5-xl	81.2
sentence-t5-large	78.4
gtr-t5-large	17.4
gtr-t5-xl	17.4
gtr-t5-base	9.5
gtr-t5-xxl	–
msmarco-bert-co-condensor	0.5
all-roberta-large-v1	35.4
all-mpnet-base-v1	25.1
all-MiniLM-L12-v1	9.7
all-MiniLM-L6-v1	3.1
multi-qa-mpnet-base-cos-v1	18.0
multi-qa-distilbert-dot-v1	1.5
stsb-mpnet-base-v2	70.2
roberta-large-nli-mean-tokens	93.6
nli-roberta-large	93.6
nli-bert-large	87.4
nli-bert-base-max-pooling	80.8
nli-bert-base-cls-pooling	83.7
facebook-dpr-ctx_encoder-single-nq-base	10.5
average_word_embeddings_levy_dependency	–
average_word_embeddings_komninos	–
average_word_embeddings_glove.840B.300d	–
average_word_embeddings_glove.6B.300d	–

Table 8: Results of the publicly available Transformer-based models in the Hugging Face sentence-transformers library.