Hy-NLI:
a Hybrid system for Natural Language Inference

Aikaterini-Lida Kalouli University of Konstanz

Richard Crouch Chegg

Valeria de Paiva Topos Institute

aikaterini-lida.kalouli@uni.kn

Abstract

Despite the advances in Natural Language Inference through the training of massive deep models, recent work has revealed the generalization difficulties of such models, which fail to perform on adversarial datasets with challenging linguistic phenomena. Such phenomena, however, can be handled well by symbolic systems. Thus, we propose Hy-NLI, a hybrid system that learns to identify an NLI pair as linguistically challenging or not. Based on that, it uses its symbolic or deep learning component, respectively, to make the final inference decision. We show how linguistically less complex cases are best solved by robust state-of-the-art models, like BERT and XLNet, while hard linguistic phenomena are best handled by our implemented symbolic engine. Our thorough evaluation shows that our hybrid system achieves state-of-the-art performance across mainstream and adversarial datasets and opens the way for further research into the hybrid direction.

1 Introduction

Natural Language Inference (NLI), the task of determining whether a premise sentence entails, contradicts or is neutral with respect to a hypothesis sentence given a specific setting, has recently seen tremendous advances. The growing availability of increasingly large datasets has enabled the training of massive deep models, pushing the state-of-the-art (SOTA) to human performance (Liu et al., 2019; Zhang et al., 2020). This performance, however, has triggered various questions: is the task now solved; is it possible to outperform humans; what do the models really understand, etc. Much work has been undertaken, with researchers detecting bias or artifacts in the training sets (Gururangan et al., 2018; Poliak et al., 2018) and "breaking" the models with adversarial datasets that expose the generalization and compositionality difficulties of the models (Glockner et al., 2018; Nie et al., 2018; Zhu et al., 2018; Dasgupta et al., 2018; Naik et al., 2018; McCoy et al., 2019; Richardson et al., 2020). This raises the question of how we can mitigate those shortcomings while taking advantage of the current advances in the field.

In this paper we propose a hybrid NLI system, which combines the strengths of a symbolic NLI engine with a deep learning (DL) model. As argued elsewhere (Lewis and Steedman, 2013; Beltagy et al., 2016), we also believe in the division of semantic labour. Distributional features are well suited for dealing with graded, fluid and robust conceptual aspects of the meaning of words, phrases, and sentences (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2019), but struggle with Boolean and contextual phenomena such as negation, modals, quantifiers, implicatives. These are phenomena to which more symbolic approaches are well suited. Thus, inference pairs of the former category that mostly rely on word/phrase similarity, and require taxonomic- and world-knowledge are considered *easy*, e.g., *The dog is walking* entails *The animal is moving.* In contrast, pairs of the latter category that require more complex inference around phenomena such as modals, negation, quantifiers, implicatives, conditionals, etc., are considered *hard*, e.g., *The boy faked the illness* contradicts *The boy was sick* (see Appendix A for more such phenomena and relevant examples). To deal with such *hard* pairs, we build a symbolic NLI engine, GKR4NLI. We hybridize this engine by exploiting the power and robustness of SOTA language representation models, which achieve high performance in the mainstream, *easy* pairs. We then train a

Proceedings of the 28th International Conference on Computational Linguistics, pages 5235–5249
Barcelona, Spain (Online), December 8-13, 2020

classifier, which determines whether the inference label of the symbolic or the DL component should be used, based on the nature of the pair, i.e., whether it involves challenging linguistic phenomena or not. Our evaluation shows that this approach achieves SOTA results across mainstream and adversarial datasets. Our contributions in the paper are three-fold: First, we implement a symbolic NLI system, GKR4NLI, and show its suitability for the task. Second, we describe how our trained classifier learns to distinguish between the *easy* and *hard* cases. Third, we show how such a hybrid setting reaches the current SOTA and propose the training of more hybrid models, able to assign each NLI problem to the most suitable solver.

2 Related Work

In recent years, two broad methodologies have been used to tackle the problems posed by NLI. On the one hand, there have been logic-based systems that involve linguistic methods such as syntactic and semantic parsing, and systems that employ a theorem prover or use *Natural Logic* and monotonicity principles (Abzianidze, 2017; Martínez-Gómez et al., 2017; Yanaka et al., 2018; Hu et al., 2020). Such systems have been largely evaluated on the SemEval-2014 version of the SICK dataset (Marelli et al., 2014b). On the other hand, NLI has been tackled with machine-learning methods in two major directions. One strand of research has trained end-to-end deep models, which transform the premise and hypothesis sentences into n-dimensional vectors and learn to classify them into one of the inference relations. Such methods range from attention architectures, e.g., Rocktäschel et al. (2016), to approaches integrating linguistic features such as syntactic parses, e.g., Chen et al. (2016), and external knowledge like WordNet, e.g., Chen et al. (2018), to models trained on multiple tasks, e.g., Liu et al. (2019), to name just a few. The other strand of work has focused on language representation models, building on the efforts of popular word (Mikolov et al., 2013; Pennington et al., 2014), sentence (Kiros et al., 2015; Bojanowski et al., 2017; Conneau et al., 2017) and contextualized embeddings, such as ELMO (Peters et al., 2018), Open GPT (Radford et al., 2018), BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019). Such representations are then efficient for supervised downstream tasks, like NLI, by only fine-tuning them on the specific task.

Although such SOTA systems achieve high performance on massive multi-domain datasets like SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018) or SciTail (Khot et al., 2018), it has repeatedly been shown that they lack the generalization and compositionality power needed to truly understand language. They are easily "broken" when presented with differently biased data (Gururangan et al., 2018; Poliak et al., 2018) or with challenging adversarial data, e.g., including semantic (Glockner et al., 2018; Zhu et al., 2018; Naik et al., 2018; Richardson et al., 2020) or structural phenomena (Dasgupta et al., 2018; Nie et al., 2018; McCoy et al., 2019). Such phenomena may include comparatives, implicatives, conditionals, coordination, negation, modals (see Appendix A for a fuller list of phenomena and relevant examples). For example, comparatives are hard because the models cannot yet efficiently capture the complex interaction between the order of the compared constituents and comparative notions like *more* or *less*. Similarly, implicatives are challenging for models because they inherently possess an implication signature which is not only unique for each predicate and thus needs to be explicitly learnt for each of them, but also undergoes changes based on how the implicative is embedded and whether it is negated. Pairs with conditionals, modals and coordination challenge the models due to the high lexical overlap between the sentences of the pair – which the models have learnt to interpret as entailment – and due to the inability of current models to exploit the notions of causality and modality.[1] Most of this "breaking NLI" work has proposed augmenting the training data with such complex phenomena and has shown that models can then successfully improve their performance. However, augmenting the training data with specific phenomena does not solve the bigger problem (Nie et al., 2018; McCoy et al., 2019). Due to the recursive nature of language, there are infinitely many ways of composing information into meaningful sentences. But each type of adversarial set can only improve performance for one specific linguistic phenomenon and already generating high-quality data for a single phenomenon is extremely costly. Indeed, Nie et al. (2018) show that training on one type of adversaries does not improve performance for other types; in fact, it might even harm the overall performance due to over-fitting. McCoy et al. (2019) also observe

[1] See relevant literature for more discussion on why models fail in such cases and explore potential heuristics of DL models through our explainable user interface presented in Kalouli et al. (2020).

that augmenting the training data with only a subset of their adversarial set enables the model to learn this subset, but does not help it solve the withheld set. Thus, we need to focus on the kind of learning process that will allow for the desired generalizations without having to include all possible constructions in the training data. One solution is to pursue a direction where such complex phenomena are treated separately by suitable symbolic components and other cases are treated by deep models. We propose training suitable hybrid models that can learn when to use which method.

Our work shows how such a hybrid approach can work in practice. Recently, work in a similar direction was conducted by Hu et al. (2020), who develop a monotonicity/Natural Logic system capable of producing entailing or contradictory sentences for a premise, in the search of the actual hypothesis. When combining their system with BERT, they improve BERT's performance on SICK by 1%. However, this slight performance boost does not reveal the real value of such a hybrid system, compared to pure deep-learning models like BERT, because it does not show its performance on adversarial sets.

3 The Hy-NLI System

The proposed system consists of a symbolic and a deep learning component, whose outputs are used for the training of the hybrid classifier. The overall architecture of Hy-NLI can be found in Figure 1. The explainability of our proposed system and an intuitive user interface are presented in Kalouli et al. (2020).

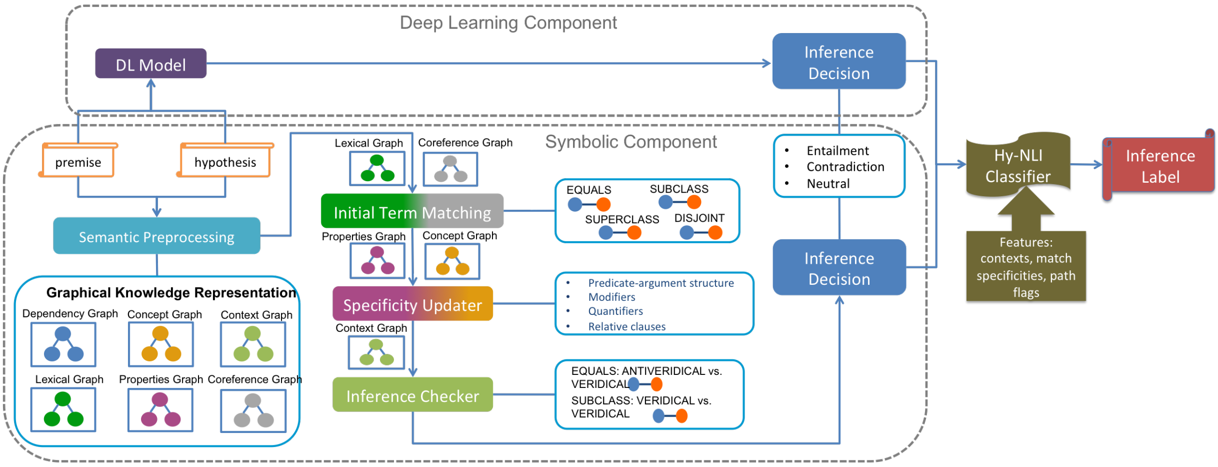


Figure 1: The overall architecture of Hy-NLI. Top: Deep learning component. Bottom: symbolic component. Right: Hy-NLI classifier.

3.1 GKR4NLI: the Symbolic Component

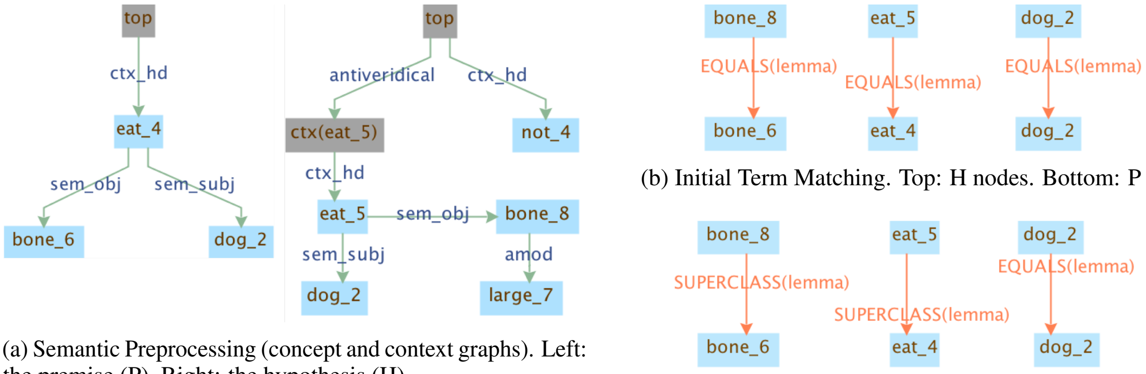
We develop GKR4NLI,[2] a symbolic inference engine which implements a version of *Natural Logic* (NL) (Van Benthem, 1986; Valencia, 1991; MacCartney and Manning, 2007; McCartney, 2009). NL seeks to determine specificity and monotonicity relations between words, phrases and sentences, i.e., determine whether the concepts of a sentence can become "more general" or "more specific" salva veritate. For example, in the sentence *a dog is eating*, *dog* can be replaced by the more general *animal* while preserving truth. GKR4NLI, fully implemented in Java, exploits an improved version of the Graphical Knowledge Representation (GKR)[3] (Kalouli and Crouch, 2018), which allows for the kind of inference mechanism we require. Briefly, the GKR parser separates the sentence information into six different graphs: the dependency graph, the concept graph, the context graph, the lexical graph, the properties graph and the coreference graph. The dependency graph is self-explanatory: it holds the dependency parse (Enhanced++ Universal Dependencies; Schuster and Manning (2016)) of the sentence. The concept graph abstracts away from the dependency graph and holds only the predicate-argument structure of the sentence, i.e., who is doing what to whom. The context graph holds the existential commitments and assertions of

[2] Available under `https://github.com/kkalouli/GKR4NLI`

[3] Available under `github.com/kkalouli/GKR_semantic_parser`

the sentence, i.e., in what worlds there are instantiations of the concepts of the concept graph. The lexical graph contains lexical information of the words of the sentence, while the property graph includes additional morpho-syntactic information about e.g., the cardinality of nouns, tense and aspect, quantifiers. Last, the coreference graph resolves coreference phenomena. Overall, an important characteristic of the representation that makes it suitable for our symbolic inference engine is its strict separation between the concept and the context graphs. For more details on GKR, refer to Kalouli and Crouch (2018). Using these representations, we implement the symbolic inference engine GKR4NLI; its overall architecture can be found on the lower level of Figure 1 and is detailed in the following. We use the pair *P: The dog is eating a bone. H: The dog is not eating a large bone* as our working example.

Stage 1: Semantic Preprocessing In the first stage, the premise (P) and hypothesis (H) are parsed to their GKR representations, each producing the six graphs outlined above. Each of the graphs is used in a different stage of the inference pipeline. Due to space limitations we only present the merged concept (blue nodes) and context (grey nodes) graphs of our working example in Figure 2a: the concept graphs capture the propositional structure (dog eat (large) bone) of the sentences, while the context graphs capture the assertions of the sentences: *eating* in P is instantiated (or the *ctx_hd* in GKR terms) in the top context (i.e., in the actual, real world), while in H it is uninstatiated (or *antiveridical* in GKR terms), which captures the fact that H says that eating of a bone by a dog does not happen.



(a) Semantic Preprocessing (concept and context graphs). Left: the premise (P). Right: the hypothesis (H)

(b) Initial Term Matching. Top: H nodes. Bottom: P nodes

(c) Specificity Update. Top: H nodes. Bottom: P nodes

Figure 2: The stages of the symbolic engine for the working example *P: The dog is eating a bone. H: The dog is not eating a large bone*.

Stage 2: Initial Term Matching We make use of the GKR lexical graphs to determine possible matches between H and P. These graphs contain the top five disambiguated WordNet (Fellbaum, 1998) senses and SUMO (Niles and Pease, 2003) concepts of each node of the concept graph, as well as their corresponding super- and subsenses/concepts, antonyms and synonyms. Based on this information and on plain word lemmas, matches between H and P are determined and assigned one of the specificity markers *equals, subclass, superclass, disjoint* – we always define the specificity of the P-term with respect to the H-term. In our working example in Figure 2b, the words *bone, eat, dog* of H (at the start nodes of the edges) match to the corresponding words in P (end nodes of the edges) solely based on their lemmas, and are all assigned the *equals* specificity. In an example where H would contain the word *animal*, the match *animal-dog* would be found based on the hypernyms of *dog*, and the specificity *subclass* would be assigned.

Stage 3: Specificity Updater The specificity of the lexical matches of the previous stage has to be updated based on the children (i.e., modifiers/arguments) that each of the P and H terms have. For instance, in our example, *bone* is equally specific to *bone*, but it is not equally specific to a *large bone* (not all bones are large). For updating the matches, we use the conceptual and properties graph of GKR.

The update process considers each match separately. For the two terms of a match H-P, the system considers whether none, both or only one of them have children in their respective concept graph. Based on that, different update rules apply.[4] For example, if both terms have no further children in their concept graphs, then the initial specificity remains unchanged (cf. *dog* in Figure 2c). On the other hand, if the H-term has additional children but the P-term does not, then H becomes more specific. Thus, in our example in Figure 2c, the specificity of the match *bone-bone* has to be revised because *bone* in H has the additional modifier *large*, while *bone* in P has no modifiers: since H becomes more specific, P becomes more general and thus the specificity *equals* is converted to *superclass* (i.e., *bone* is more general (superclass) than *large bone*). Similarly, the specificity of the match *eat-eat* has to be adapted based on the children of its terms: the children *dog-dog* and their specificity have no effect on the specificity of *eat-eat* because an equal match has no effect on an equal match. In contrast, the *superclass* specificity of the children *bone-bone* modifies the specificity of *eat-eat* by projecting up, and converts it to *superclass*.

The matching process described so far will not always be valid because the matched terms might have different, incompatible relations in the sentences. For instance, in our working example, the matched children *dog-dog* of the match *eat-eat* are compatible and valid because they both stand in the *sem_subj* relation to *eat* (see Figure 2a). However, in a pair like *P: The dog is chasing the cat. H: The cat is chasing the dog*, the matched children *dog-dog* of the match *chase-chase* are not compatible because *dog* is the *sem_subj* of P but the *sem_obj* of H. So, since two children may match no matter their relation to their head, and thus produce an invalid match, the matches of each pair are assigned a flag based on whether they are more often associated with an (E)ntailment, a (C)ontradiction or a (N)eutral relation. To find such associations, we use association rule mining (Agrawal et al., 1993), a rule-based machine learning method for discovering interesting relations between items in large databases. Specifically, we use the fpgrowth algorithm (Han et al., 2000): we run an annotated inference training set (see Section 4 for more details on the set) through the first 3 stages of the pipeline and extract all possible matches for each pair and map them to their semantic relations, e.g., in our working example, the matches *dog-dog* and *bone-bone* are mapped to their semantic relations {*sem_subj – sem_subj*} and {*sem_obj – sem_obj*}, respectively. A balanced subset of these relations-combinations along with the inference label of each pair is used as input (itemsets) for the association algorithm, as shown in Table 1. The algorithm then learns rules for which relations-combinations are mostly associated with what label. We manually verify the rules with the highest confidence and bootstrap them in the current stage to act like the flagging mechanism for the pair. For example, the algorithm finds that the combination of the matched semantic relations {*sem_subj – sem_obj*} and {*sem_obj – sem_subj*} in one sentence is mostly found in Cs; we verify this rule, which allows us to account for pairs like *P: The dog is chasing the cat. H: The cat is chasing the dog* or *P: The fish is following the turtle. H: The turtle is following the fish.* Similarly, the algorithm finds that the combination of the matched relations *amod,sem_subj – amod,nmod,sem_subj* with the other items shown in Pair 2 of Table 1 is mostly found in neutral pairs: in P, *asian* is the *amod* (adjectival modifier) of *people* and *people* is the *sem_subj* of *eat*. In H, *asian* is the *amod* of *restaurant*, *restaurant* is an *nmod* (nominal modifier) of *people* and *people* is the *sem_subj* of *eat*. We can then verify that such a relations-combination should indeed be flagged as neutral. Currently, there are more than 20 verified rules included in the system, with the possibility of extension.[5]

Once all matches have been updated, we still need to deal with determiners/quantifiers. Quantifiers can flip the specificity of a match, according to the monotonicity principles (van Benthem, 2008; Valencia, 1991; McCartney, 2009). For instance, *bone* is more general that *large bone*, but *every bone* is more specific than *every large bone*, i.e. *every large bone* does not entail *every bone*. The determiner/quantifier information is provided in the GKR properties graph.

Inference Checker After all the H-P matches have been updated, the inference relation is determined based on the GKR context graphs and the determined specificities. First, we look for contradictions coming from opposing contexts. A contradiction requires an instantiated term (termed *veridical* or *ctx_hd*

[4] See Appendix B for the full update rules.

[5] Note that this method will not always learn accurate associations. For example, in a pair like *P: The boy met the girl. H : The girl met the boy*, the combination {*sem_subj – sem_obj*}, {*sem_obj – sem_subj*} should not lead to a contradiction but to an entailment due to the nature of the predicate *meet*. However, attempting to learn lexicalized associations, i.e., including the specific lexical items and/or their Levin (Levin, 1993) class, did not lead to reliable associations due to the high sparsity of each relation-combination. Richer corpora are needed for better (lexicalized) association mining.

Transactions	Itemsets	Example
Pair 1	{sem_subj – sem_obj} {sem_obj – sem_subj} {C}	P: The turtle is following the fish. H: The fish is following the turtle

Pair 2	{sem_subj – sem_subj} {nmod – nmod,sem_subj}
	{amod,sem_subj – amod,nmod,sem_subj} {N}
		P: Asian people are eating at a restaurant.
		H: People in an Asian restaurant are eating.

Pair 3	{sem_subj – sem_subj} {amod,nmod-nmod}
	{amod –} {E}
		P: The cat is playing passionately with a watermelon.
		H: The cat is playing with a watermelon

Table 1: Sample itemsets (input) for the fpgrowth algorithm. The left-side of the dash represents P and the right-side H. No relation before/after the dash means that there was no match for this term.

within GKR) being matched with an uninstantiated term (*antiveridical*) in the same context. For example, if the H-term is instantiated and more or equally specific than the uninstantiated P-term, then there is a contradiction, as in *P: No man is walking. H: A (young) man is walking.* In contrast, if the H-term is uninstantiated and more or equally specific than the instantiated P-term, we cannot determine the relation, as in our working example: from *A dog is eating a bone* we cannot infer whether the dog is eating a large bone or not. Similar rules apply for contradictions coming from disjoint relations. For instance, if the root node of the H-graph has a disjoint match in the P-graph and both terms have the same instantiability and the path of the match is not associated with a contradiction flag, there is a contradiction, as in the example *P: The man is carrying a big bag. H: The man is carrying a small bag*: the root node *carry* of H has the match *carry* in P, both are instantiated in the top context, no contradiction flag is assigned to them, but they are in a disjoint relation due to their disjoint modifiers *big* and *small*. Similarly, we find entailments. For example, if the root node of H-graph has a match in the P-graph, the match is equally or more specific, both terms are instantiated and the path of the match is not associated with a contradiction or neutral flag, there is an entailment, as in the pair *P: The dog is eating. H: The animal is eating.* If none of the rules apply, the relation cannot be determined and defaults to neutral.

3.2 The Deep Learning Component

For the deep learning (DL) component we choose to experiment with two state-of-the-art language representation models, BERT (Devlin et al., 2019) (base, uncased) and XLNet (Yang et al., 2019) (base), which we fine-tune for our task. Details on the train and test sets are given in the next section. We use the *HuggingFace* implementation of the models[6] and we fine-tune the parameters suggested by the authors: batch size, learning rate and number of epochs. Our best performing models use a batch size of 32, learning rate of 2e-5 and 3 epochs. The trained model can classify an input inference pair into E, C or N.

3.3 The Hybrid Classifier

Each of the two described components makes an inference decision for a given pair, as shown in Figure 1. What we need to know is which of the two decisions we should trust, if any. If we go with the DL decision, we will probably do well on the mainstream (easy) pairs but performance will suffer for the adversarial ones, as described in Sections 1 and 2. If we go with the decision of the symbolic component, we might miss some of the easier cases due to lower robustness of the system, but we will have high performance on the adversarial cases. So, if we can determine whether a pair is *hard* or *easy*, i.e., if it contains phenomena that require deeper semantic analysis like modals, negations, quantifiers, implicatives, factives, etc., we can choose to trust the component we know to work best in each case. Most of these harder phenomena cannot be traced solely based on surface forms, e.g., even if modals can be captured through a short list of words, the semantic implications that actually make them complex cannot be modeled so straightforwardly. But the GKR architecture lends itself to the extraction of such information: the GKR context graphs capture exactly the complex implications and assertions that come from such phenomena. Thus, to distinguish between *hard* and *easy* pairs, a classifier only needs to learn which combinations of such implications and of match specificities are indeed complex and which are easier.

To learn that, the classifier is given a fake task: it learns which component of our system delivers the correct inference label for a given pair. By learning this, the classifier indirectly learns whether the pair

[6] Available under `github.com/huggingface/transformers`

is *hard* or *easy*: if the symbolic component is right, the pair is probably *hard*; if the DL component is right, the pair is probably *easy*[7]; if both components can get the inference right, then the pair does not require any special treatment; if none of the components gets it right, then no claims about the nature of the pair can be made. Thus, to build the training set for our classifier, for each H-P pair, we convert the following information to features: the specificity relations (*equals, subclass, superclass, disjoint*) and the instantiability (*veridical, antiveridical, averidical*) of all matches of the pair and the path flags (*entail_flag, contra_flag, neutral_flag*) of the matches. So, our working example would be encoded by the features *veridical, antiveridical, equals, superclass* due to the instantiabilities of *eat* in P and H (see Figure 2a) and the updated specificities of the matches (see Figure 2c). As learning target, we annotate each pair with the component that assigns to it the correct inference label (based on the gold labels provided by the training set): the symbolic one (S), the DL one or both of them (B). If none of the components delivers the right label, this pair is left out of the classifier's training to reduce training noise (since these are only a few cases). We experiment with various classifiers and best performing is a Multi-Layer Perceptron (MLP) classifier with 8 hidden layers, the ReLU activation function (Hahnloser et al., 2000), Adam solver for weight optimization (Kingma and Ba, 2014), adaptive learning rate, L2 penalty at 0.01, learning rate initialization at 0.01 and maximum iterations of 1000.[8] The classifier learns one of the labels S, DL or B (3-way classification). In the testing phase, each pair is classified as one of S, DL or B and then mapped to the respective label: if classified as S or DL, the symbolic or the DL inference label are used, respectively; if classified as B, then either one of S or DL could be chosen: to increase robustness the label of the DL component is preferred . After this classification, a final unique inference relation is assigned to each pair. Table 2 gives a glimpse into the output of the classifier.

It should be noted that the proposed hybrid approach is different from an approach which simply uses a DL model whenever the symbolic engine fails or produces a neutral label (e.g., Hu et al. (2020)). In such approaches, the core of the approach is the symbolic engine and the DL model is just a fall-back strategy. In Hy-NLI, however, both components have an equal standing and there is an apriori decision for which component should be trusted. With this, we can fully exploit the robustness and power of DL models. Additionally, Hy-NLI can be used to reliably label unannotated data. In contrast, approaches that employ the DL fall-back strategy can mainly be exploited in settings where gold labels are available and thus the fall-back strategy can be triggered.

Pair/Label	**Symbolic**	**DL**	**Hybrid (S, DL, B)**	**Mapped**	**Gold**
P: The artist encouraged the secretary. H: The secretary encouraged the artist.	C	E	S	C	C
P: A man is running. H: A man is standing still.	N	C	DL	C	C
P: A boy is happily playing the piano. H: A white bird is landing swiftly in the water.	N	N	B	N	N

Table 2: Sample output of the Hy-NLI classifier and its mapping to real inference labels.

4 Evaluation

4.1 Datasets

For evaluation we choose three different datasets: SICK (Marelli et al., 2014b), the set of Dasgupta et al. (2018) (DAS) and HANS (McCoy et al., 2019). SICK is considered an easy dataset, where DL approaches do well. DAS is one of the adversarial datasets, where DL approaches struggle. HANS lies squarely in the middle: half of it can be easily solved by DL methods and half of it is hard for such models.

SICK SICK is an English corpus of almost 10,000 pairs, annotated for the similarity and for the inference relation between the sentences of each pair. The corpus was created from captions of pictures

[7] Recall that "easy" refers to sentences not involving any semantically complex phenomena but this does not mean that the inference is an easy inference. A pair like *P: Chicago Bulls won the game. H: A basketball game took place* is not semantically complex, but still requires a good deal of world-knowledge to get the inference right.

[8] Available under `https://github.com/kkalouli/Hy-NLI`

talking about daily activities and non-abstract entities. For guidelines the annotators were given one example for each inference type and no specific directions. This process caused much confusion (Marelli et al., 2014b), especially since it did not resolve event and entity coreference issues. So, a pair like *A woman is carrying a bag* and *A woman is not carrying a bag* ended up labeled as N in its A → B direction, but as C in the B → A direction, because the indefinite *a* could be referring to two different women or not, depending on how one defines the task. To mitigate this shortcoming, we work with a corrected version of SICK made available by Kalouli et al. (2017).[9] The correction work has only revised the pairs originally labeled as E or C, i.e., around 1/3 of the corpus. Although we also report performance on the original SICK, we focus on the corrected version, which naturally leads to better results. For all purposes, we use the SemEval 2014 version of SICK (Marelli et al., 2014a) with the corresponding test and train splits: the train split is used for learning the association rules (Section 3.1), fine-tuning the DL models (Section 3.2) and training the Hy-NLI classifier (Section 3.3), and the test split is used for evaluation.

We do not evaluate on the commonly-used benchmarks of SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) after careful inspection of the corpora. Specifically, we investigated whether these corpora suffer from the same problems as the original SICK corpus. Although the event coreference weakness was specifically targeted in these corpora, there still seems to be confusion between the relations of C and N. Contradictions need a common reference background, as argued in Zaenen et al. (2005) and de Marneffe et al. (2008), but this is not what happens with contradictions in these corpora: a pair like *Two women are embracing while holding to go packages* and *The men are fighting outside a deli* is labeled as C, although there is no coreference whatsoever between them. This confusion was also discussed in a recent experiment by Kalouli et al. (2019) and in the work of McCoy et al. (2019). Since this notion of C and N does not correspond to our definitions of C and N, and since the annotations of these corpora seem inconsistent, we do not learn or test on these corpora.

DAS Dasgupta et al. (2018) create an NLI set of 40,000 pairs that cannot be solved only with word-level knowledge but instead involve more complex phenomena. The set contains three different phenomena between sentences: the *same* type (P and H only differ in the order of the words), the *more-less* type (comparison sentences with more/less) and the *not* type (P and H differ by whether they contain the word 'not'). An example combining the two latter types is *P: The woman is more cheerful than the man. H: The man is not less cheerful than the woman*, labeled as C. They train a classifier on SNLI, using the InferSent (Conneau et al., 2017) embeddings, and find that the performance on all of their created sets barely reaches 50%, when there is no explicit training on these phenomena. For the current work, we use the entire 40,000 dataset for testing; no portion of this dataset is used for learning association rules, fine-tuning the DL model or training the Hy-NLI classifier, since we want to test how much we can achieve with the hybrid approach, especially GKR4NLI, when such phenomena are not explicitly in the train set.

HANS McCoy et al. (2019) create a hard NLI set of 30,000 pairs with syntactic heuristics: the lexical overlap heuristic, the subsequence heuristic and the constituent heuristic. These three broad categories include 5 phenomena each, from passives and relative clauses to coordination and embedded verbs under factives, etc. For example, the set contains pairs with embedded verbs like *P: The lawyer knew that the judges shouted. H: The judges shouted* or pairs with coordination *P: The artist and the student called the judge. H: The student called the judge.*, etc. McCoy et al. (2019) collapse the labels C and N to *non-entailment* after training, as they also find that the annotation of the two is not clear-cut in in the training corpus.[10] They train four different SOTA models on MNLI and test on their adversarial set, showing that performance is high when the correct answer is in line with the hypothesized heuristic (one half of the corpus), but drops under chance when the heuristic leads to an incorrect prediction (other half of the corpus). For our evaluation, we use the entire 30,000 dataset, again without any portion used for any kind of training.

[9] Available under `github.com/kkalouli/SICK-processing`

[10] They also experiment with collapsing the labels before training, but the results are the same.

4.2 Results

In Table 3 we compare SOTA logic-based systems, end-to-end DL models like ESIM (Chen et al., 2016) and language representation models like BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019), our symbolic engine GKR4NLI and our hybrid Hy-NLI system. The *All* column shows the average accuracy of each setting across datasets.[11] The results confirm previous published findings and demonstrate the value of the proposed approach. Concerning SICK, BERT and XLNet achieve SOTA performance at 85-86% (comparable to BERT's and XLNet's performance on SNLI and MNLI), while the logic-based ccg2lambda and the symbolic GKR4NLI reach 73% and 78% accuracy, respectively. Concerning the adversarial sets, the DL models' performance falls short and they only achieve majority-baseline performance – consistent with the findings of Dasgupta et al. (2018) and McCoy et al. (2019). In contrast, on these sets GKR4NLI shows performance above 70% and 90% for HANS and DAS, respectively. Across datasets and thus across linguistic complexity, the DL models barely reach an average performance of 63%, while GKR4NLI achieves 80.3%. ccg2lambda is also competitive in the HANS set, but fails to perform in the DAS set, thus achieving an average accuracy of 62.1%. By combining the best of both worlds, Hy-NLI with BERT is able to achieve an average accuracy of 81.5% across datasets, almost 18% higher than the pure DL models and 1.5% to 19% higher than the purely logical systems of GKR4NLI and Yanaka et al. (2018), respectively. With this, it approaches the ceiling average performance of 83%, which suggests that for each dataset it almost achieves the performance of the best component. On the other hand, Hy-NLI with XLNet does not deliver as good results. Interestingly, even when comparing the pure DL models, XLNet delivers worse results than BERT in two of the three datasets, although the differences are negligible to a large extent. This could be due to the fine-tuning process being more optimized for BERT than XLNet or even to the models' architecture, which favors one kind of data over others. Hy-NLI with XLNet struggles to reach the best performance, especially for the hard datasets DAS and HANS.

Method	**SICK**	**DAS**	**HANS**	**All**
Baselines				
hypothesis-only baseline (Poliak et al., 2017)	56.8	-	-	-
majority baseline	56.8	50	50	52.2
Logic-based systems				
ccg2lambda (Yanaka et al., 2018)	73.3	32.8	80.4*	62.1
MonaLog (Hu et al., 2020)	52.5	16	50	39.5
DL/RL systems				
InferSent (Kiros et al., 2015)	-	48.6	-	-
ESIM (Chen et al., 2016)	80.7	50.3	47.6	59.5
BERT (Devlin et al., 2019)	86.5*	50.2	47.5	61.4
XLNet (Yang et al., 2019)	85.8	49.9	53.6	63.1
Hy-NLI (this work)				
GKR4NLI (symbolic component)	78.5	90.8*	71.7	80.3
Hy-NLI (with BERT)	84.8	90.8	68.9	**81.5**
Hy-NLI (with XLNET)	83	72.6	53.5	69.7
Ceiling performance (best components's performance)	86.5	90.8	71.7	83

Table 3: Accuracy on the three datasets. *All* shows the average accuracy of each setting across datasets. *figures are the best performances for each dataset. Bolded figure is the best performance across datasets.

5 Discussion

Overall, the reported results show the efficiency of the proposed approach. By utilizing general features about the semantic nature of the pair, Hy-NLI becomes competitive to the average ceiling performance across datasets.[12] The fact that the system does not reach the ceiling performance in two of the three datasets suggests that the training of the classifier did not include several of the feature combinations. SICK, a training corpus of simple inferences, does not contain a large variety of feature combinations

[11] Note that the results reported for SICK concern its corrected version by Kalouli et al. (2017) and do not include any manual pre/post-processing and might thus differ from the original numbers reported by the creators of the respective systems.

[12] The ceiling performance refers to the performance of the best component of Hy-NLI on each dataset, and not of the best system in general. The computation of the latter is pointless for systems not offering hybridization methods.

of complex phenomena. Thus, the classifier could only learn a limited number of types of complex inferences, which is reflected in the lower performance for HANS (which combines *easy* and *hard* cases, as discussed). On the other hand, the sampling process might also have disadvantaged the learning of *easy* patterns because the pairs on which only the DL component gave the correct label had to be severely downsampled to match the low number of pairs on which only GKR4NLI gave the correct label (again, due to the nature of the SICK corpus that mainly contains *easy* pairs). Thus, this could have led to the performance difference for SICK, where Hy-NLI's performance misses the best performance by 1.7%. In fact, an error analysis showed that the errors originate either from the hybrid classifier predicting the wrong component or the classifier predicting the correct component but the component itself predicting the wrong label. In the former case, for a pair like *P: Several people are in front of a building which is covered by colors. H: Several people are in front of a colorful building*, Hy-NLI predicted S and was thus mapped to the neutral label of GKR4NLI; however, given the required robustness of *colorful – covered by colors*, the better label would have been DL which would have indeed mapped to entailment. In the latter case, there are pairs like *P: The player is missing the basket and a crowd is in background. H: The player is dunking the basketball into the net and a crowd is in background*, where Hy-NLI correctly predicts DL (no complex phenomenon involved, only lexical semantics), but the DL component incorrectly assigns the neutral label as it fails to recognize the contradiction. This shows that there is potential in improving all parts of the system: the hybrid classifier as such but also its distinct components. We can also observe that the average performance of Hy-NLI across datasets is higher only by a small margin than the average performance of GKR4NLI. This is not so puzzling considering that Hy-NLI achieves higher performance than GKR4NLI on SICK but lower performance on HANS, bringing the two average performances closer.

Hy-NLI achieves its goal to combine the strengths of very different components and is able to perform across the board. By considering what types of inference each kind of component is best at, the system can successfully implement the idea of using the most suitable solver for a given problem. By exploiting features about the semantic nature of a pair, it can determine which pairs should be best handled by which component. The features that Hy-NLI exploits are generalizable semantic properties and not specific sentence combinations, so that the augmentation of the training data with further properties combinations should give a direct boost to the performance. In fact, given the small number of features used for training, this kind of classifier should be able to perfectly learn to classify the pairs, if presented with all combinations of the few used generic features.

Training and testing on the original SICK (not the corrected version) does not change the overall picture: BERT's accuracy remains the same for all three datasets and the accuracy of XLNet improves to 88% for SICK but remains the same for HANS and DAS. The accuracy of GKR4NLI unsurprisingly drops to 76.3% for SICK but remains stable for the hard sets. This means that the slight differences balance each other out and the overall performance of Hy-NLI remains the same.

The reader might also wonder what happens with pairs where both DL robustness/world-knowledge and symbolic reasoning are required. Currently, NLI datasets contain pairs that are either linguistically complex or lexically/world-knowledge-wise complicated. Thus, this otherwise very realistic situation does not occur and Hy-NLI does not need to deal with it. However, Hy-NLI has the potential of doing so by extending the initial term matching stage of GKR4NLI: the matching process could be enhanced with embeddings, thus gaining on robustness and informativity. With such an improvement, dual cases, where symbolic reasoning and DL robustness are required, could still be best handled by GKR4NLI.

6 Conclusion

This paper presented a hybrid NLI system that marries up the strengths of a symbolic and a DL component to close the gap between the performance of SOTA models on mainstream and adversarial datasets. The proposed system achieves SOTA results across the tested datasets, highlighting the advantages of the approach. Thus, we propose concentrating on such hybrid settings, able to learn which technology is most suitable for the problem at hand. Future work on Hy-NLI involves determining how the properties of each component are used and hence how to improve them. The modularity of the system allows each component to be improved independently by focusing on the problems each component is good at.

Appendix A: Examples of complex linguistic phenomena where DL models struggle

Phenomenon	Example	Adversarial Set
comparatives	P: The waiter is less disgusted than the teacher. H: The teacher is more disgusted than the waiter.	Dasgupta et al. (2018)
implicatives	P: The judge believed the tourist arrived. H: The judge believed the tourist.	McCoy et al. (2019)
conditionals	P: If the judge encouraged the managers, the lawyers supported the doctors. H: The judge encouraged the managers.	McCoy et al. (2019)
coordination	P: The secretary and the managers saw the actor. H: The secretary saw the managers.	McCoy et al. (2019)
negation	P: Enthusiasm for Disney's Broadway production of The Lion King dwindles. H: The broadway production of The Lion King is no longer enthusiastically attended.	Naik et al. (2018)
modals	P: And, could it not result in a decline in Postal Service volumes across–the–board? H: There may not be a decline in Postal Service volumes across–the–board.	Naik et al. (2018)
word-order scrambling	P: A woman is pulling a child on a sled in the snow. H: A child is pulling a woman on a sled in the snow.	Nie et al. (2018)
passivization	P: Harley asked Abigail to bake some muffins. H: Abigail is asked to bake some muffins.	Zhu et al. (2018)

Table 4: Examples of complex linguistic phenomena where DL models struggle. This list is not meant to be exhaustive; see relevant literature for more cases.

Appendix B: Update rules for the specificity update stage of the GKR4NLI pipeline

Initial Specificity	Updated Specificity	
	H has dependents, P does not ⇒ H more specific	P has dependents, H does not ⇒ P more specific
equal	superclass	subclass
subclass	none	subclass
superclass	superclass	none
disjoint	none	none
none	none	none

Table 5: Updated specificity of a match, when only one term has dependents.

H-P Specificity	Hdependent-Pdependent Specificity				
	equals	subclass	superclass	disjoint	none
equals	equals	subclass	superclass	disjoint	none
subclass	subclass	subclass	none	disjoint	none
superclass	superclass	none	superclass	disjoint	none
disjoint	disjoint	none	none	disjoint	none
none	none	none	none	none	none

Table 6: The computation of the updated specificity of an H-P match, when both terms have dependents.

References

Lasha Abzianidze. 2017. LangPro: Natural language theorem prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 115–120, Copenhagen, Denmark, September. Association for Computational Linguistics.

Rakesh Agrawal, Tomasz Imieliundefinedski, and Arun Swami. 1993. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June.

Iz Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. 2016. Representing Meaning with a Combination of Logical and Distributional Models. *Computational Linguistics*, 42(4):763–808.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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, September. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia, July. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September. 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.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1047, Columbus, Ohio, June. 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, June. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

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. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Richard H. R. Hahnloser, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, and H. Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951.

Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 1–12, New York, NY, USA. ACM.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a Lightweight System for Natural Language Inference Based on Monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 284–293, New York, New York, January. Association for Computational Linguistics.

Aikaterini-Lida Kalouli and Richard Crouch. 2018. GKR: the Graphical Knowledge Representation for semantic parsing. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 27–37, New Orleans, Louisiana. Association for Computational Linguistics.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2017. Correcting Contradictions. In *Proceedings of Computing Natural Language Inference (CONLI) Workshop, 19 September 2017*.

Aikaterini-Lida Kalouli, Annebeth Buis, Livy Real, Martha Palmer, and Valeria dePaiva. 2019. Explaining simple natural language inference. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 132–143, Florence, Italy, August. Association for Computational Linguistics.

Aikaterini-Lida Kalouli, Rita Sevastjanova, Richard Crouch, Valeria de Paiva, and Mennatallah El-Assady. 2020. XplaiNLI: Explainable Natural Language Inference through Visual Analytics. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, COLING '20. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTaiL: A textual Entailment Dataset from Science Question Answering. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Mike Lewis and Mark Steedman. 2013. Combined Distributional and Logical Semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2007. Natural Logic for Textual Inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague, June. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, August. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 710–720, Valencia, Spain, April. Association for Computational Linguistics.

Bill McCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University, Stanford, CA, USA. AAI3364139.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR*.

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, August. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2018. Analyzing compositionality-sensitivity of NLI models. *CoRR*, abs/1811.07033.

Ian Niles and Adam Pease. 2003. Linking Lexicons and Ontologies: Mapping Wordnet to the Suggested Upper Merged Ontology. In *Proceedings of the IEEE International Conference on Information and Knowledge Engineering*, pages 412–416.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew 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, June. Association for Computational Linguistics.

Adam Poliak, Pushpendre Rastogi, M. Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 503–508, Valencia, Spain, April. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing Natural Language Inference Models through Semantic Fragments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8713–8721. AAAI Press.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*, San Juan, Puerto Rico.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Victor Sánchez Valencia. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Johan Van Benthem. 1986. *Essays in logical semantics*. Springer.

Johan van Benthem. 2008. A brief history of natural logic. In M. Chakraborty, B. Lowe, M. Nath Mitra, and S. Sarukki, editors, *Logic, Navya-Nyaya and Applications: Homage to Bimal Matilal*. College Publications.

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, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki. 2018. Acquisition of phrase correspondences using natural deduction proofs. 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 756–766, New Orleans, Louisiana, June. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR*, abs/1906.08237.

Annie Zaenen, Lauri Karttunen, and Richard Crouch. 2005. Local textual inference: can it be defined or circumscribed? In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pages 31–36. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.

Xunjie Zhu, Tingfeng Li, and Gerard de Melo. 2018. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637, Melbourne, Australia, July. Association for Computational Linguistics.