Entailment, Contradiction, or Neutrality? An Exercise in Monoton(icit)y

Shiyi Butter s.y.j.butter@students.uu.nl
Lara Gierschmann l.k.gierschmann@students.uu.nl
Jeroen Spaans j.p.spaans@students.uu.nl
Yulia Terzieva y.i.terzieva@students.uu.nl

Abstract

This study examines the extent to which rule-based systems can model human reasoning in the context of Natural Language Inference. Inspired by monotonicity and the problems of large language models (e.g. environmental impact), we use the Stanford Natural Language Inference dataset to analyze crowd workers' reasoning and mimic it. We explore a baseline system and a tree system and their combination, achieving 44.9% accuracy. We acknowledge the shortcomings of our approach and of the tools we use and discuss them in detail.

1 Introduction

Recently developed language models such as OpenAI's ChatGPT and Meta's Galactica have been the talk of the year, however, many drawbacks of the approaches those models adopt are left outside of the spotlight. First and foremost, looking at the size of the models, as measured by the size of the training data and the number of parameters, environmental risks are identified (Bender et al.). It is estimated that training a single deep learning system leaves an enormous carbon footprint comparable to five vehicles for personal use based on data from 2019 (Strubell et al.), which is only increasing with the current state-of-the-art models. The difficulty of understanding what the models encode and what the training data holds is scaling together with the environmental impact. A new line of thought, presented by Bender et al., encourages researchers to direct their attention toward approaches that do not depend on having large language models. We believe one such approach could be a rule-based Natural Language Inference system.

Natural Language Inference (NLI), also known as Recognizing Textual Entailment, is a three-way classification task. A dataset developed for this is the Stanford Natural Language Inference (SNLI) dataset (Bowman et al.). Given two human-written sentences an inference relation – *entailment, contradiction* or *neutral* must be assigned.

In this paper, we explore the extent to which we can model the crowd workers' reasoning using a rule-based system. We have developed a baseline system and a monotonicity-based system and the combination of those two yields an accuracy of 44.9%.

Using rule-based systems to model the meaning of natural language is impractical, due to the vast amount of manual work in hand-crafting the rules. However, this type of research sheds light on the reasoning humans have when dealing with tasks such as NLI, which can be of great contribution to future research.

2 Methods

The rule-based model is developed based on a subset of the SNLI dataset. For the training, we used the first ten thousand problems from the training set, each being a pair of premise and hypothesis. The distribution of labels was maintained equal.

2.1 Baseline

The most fundamental way of comparing sentences is to determine whether the main verbs and subjects of both sentences are the same or of similar meaning, as this has the largest impact on the meaning of the whole sentence. To obtain those main verbs and subjects of each sentence, we have used the roots of the dependency trees generated with spaCy (an open-source Python library) (Honnibal and Montani, 2017). This is possible because the root takes on a verb POS tag in 81% of the sentences and a subject tag in 18% of the sentences. To facilitate accurate comparison both root tags have to be of the same type. Given this, if the POS tag is not the same (e.g., the premise's root has a

verb tag and the hypothesis's root a subject one) we instead select the word to which the verb root is pointing to, and compare that to the subject root. This is possible as in most cases the verb root is pointing to a word with a POS tag subject. For example, the sentence pair *A dog standing near snow looking at water.* and *A cat is laying on the couch.* has roots *dog* and *laying*, but after the procedure, we get roots *dog* and *cat*. Figure 2 illustrates this example.

To increase the accuracy of this basic root comparison, we select the word to which the roots are pointing (hereafter children) and compare them if they have the same POS tag.

In the cases that one of the sentences is notably longer (implemented as 1.15 times longer), we select the grandchildren of the root and also compare those with the children of the shorter sentence's root. This is because when one sentence is longer than the other, a word such as a noun or a verb is connected to more adjectives or adverbs, or another prepositional clause is added. In these cases, the children of the shorter sentence might be related to the grandchildren of the longer sentence. By testing different variables for sentence length (starting with 1.5 times longer for the long sentence), we found that comparing the grandchildren with the children achieves the highest accuracy when the long sentence is at least 1.15 times longer than the short sentence.

Each comparison results in a single label – *entailment, contradiction* or *neutral*. Based on the combination of those labels we determine whether the sentences (premise and hypothesis) are in an *entailment, contradiction* or *neutral* relation. If there is a single *contradiction* label we conclude *contradiction*, otherwise we look at the root pair comparison – if the label of that pair is *entailment* we conclude the *entailment*, otherwise we conclude the majority label.

An example of this is illustrated in Figure 3. The roots of both sentences are *playing*. In both sentences, the children of the root are *woman, is, violin*. We do an element-wise comparison between them to see what pairs have the same POS tag and end up with *[woman, woman]*, *[woman, violin]*, *[is, is]*, *[violin, woman]* and *[violin, violin]*. Although it is unnatural to compare *violin* with *woman*, the baseline system does so because the POS tag of both is a noun. The labels for each pair are respectively *['entailment', 'entailment', 'neutral', 'entailment', 'neutral', 'entailment']*. Following the rules given in the section above, the overall label given to the premise and hypothesis is *entailment*.

2.1.1 Word comparison

The comparison of the generated word pairs is based on several essential lexical relations. For obtaining those relations we have used WordNet (Miller, 1998). The comparison is done on the lemmatized words. First and foremost, we determine whether the words are antonyms, in which case we label the pair as *contradiction*. When determining whether the words are antonyms, we do not only consider adjectives, but also relational antonyms, such as "walk", and "ride", based on WordNet. Contrary, if two words are synonyms we label them *entailment*. However, WordNet considers only strict relations and it is lacking completeness. For example, in the context of SNLI, words such as 'cat' and 'dog' are antonyms but WordNet would not flag these as such. Due to this limitation, we also added hard-coded pairs containing the most common antonyms and synonyms in the dataset. To obtain these words, we first calculated the most common word pairs. For instance, the most common word pair for nouns is 'man' and 'shirt' with a frequency of 449. This means that 449 times, 'man' is in the premise and 'shirt' is in the hypothesis. Next, we looked at the gold label and counted the number of contradictions for each word pair. If 70% or more of the labels were contradictions, the word pair was added to the antonyms. In this case, the word pair 'man' and 'shirt' was not included in antonyms since 33.63% of the labels was a contradiction. This procedure is the same for finding word pairs for entailment and neutral problems.

If the words are not antonyms and not synonyms, we look if the first word is a hyponym or a hypernym of the second word. If a premise word is a hyponym of a hypothesis word, then the premise word is more specific than the hypothesis word. Hence, we label them as *entailment*. Suppose 'dog' is a premise word and 'animal' is the hypothesis word, then 'dog' is a hyponym of 'animal' and therefore it is labeled as *entailment*. Suppose 'animal' is the premise word and 'dog' is the hypothesis word, then 'animal' is a hypernym of 'dog'. Thus, 'animal' is more general than 'dog' and therefore they are labeled as *neutral*.

Lastly, we check if two words are co-hyponyms, i.e., when two words share the same hypernym(s). For example, 'mountain' and 'beach' share 'geolo-

gical formation' as a hypernym. If a co-hyponym is found, we label the words as *contradiction*. This function is implemented in two different ways. The baseline co_hyponym function compares the two direct hypernym sets. If the two sets have a hypernym in common, the two words are labeled as *contradiction*. The tree co_hyponym function calculates the shortest path distance for two words by determining the shortest path in the hypernym/hyponym hierarchy that connects them. The two words are labeled as *contradiction* if the distance is smaller than or equal to 3. This distance constraint is set to limit the influence of two words sharing a common hypernym that is high up in the hierarchy. For instance, 'table' and 'car' have 'entity' in common with the shortest distance of 15. The reason for implementing two different different co_hyponym functions is that, as will be explained in section 2.3, there are problems for which trees can be made and problems for which there cannot; the baseline co_hyponym function performs better on problems that do not allow for the generation of trees, and the tree co_hyponym function performs better on those problems that do.

2.2 Referent Trees

To facilitate the comparison between premise and hypothesis, the sentences were translated into trees in which the nodes represent referents of the sentence (i.e. the entities or actions tokens refer to) and the edges represent syntactic relations between these referents. Abstracting to the referents of each sentence makes it easier to determine which tokens to compare.

As an example, let us consider the sentence 'A man in a red life vest is riding in a canoe.'. This sentence refers to three separate entities and a single action. Relating these to node names E_x for entities and V_x for actions we have the following, also found in figure 1a:

E_0**:** a man

E_1**:** a red life vest

E_2**:** a canoe

V_0**:** is riding

E_0's man is the subject of V_0's riding. To represent this, we place an arc labelled *subject* from V_0 to E_0. Likewise, for sentences with an object of a verb, an arc labelled *object* is placed from the action node to the entity node. There is one more type of relation modelled in referent trees: the preposition. Our example sentence contains two prepositions; the man is *in* the life vest, and he is riding *in* the canoe. We model this with arcs from E_0 to E_1 and from V_0 to E_2, both labelled *preposition*.

Since spaCy (which is used for dependency parsing) finds a verb to be the root of the majority of sentences in the data, we follow suit for our referent trees and have a root node point to the corresponding action node. In this example, this means node V_0, corresponding to the spaCy dependency root *riding*, is related to by a root node. Putting this all together gives us the left tree in figure 1b. The right tree in figure 1b, described in figure 1c, is for an example hypothesis to this example's premise. Naturally, not all sentences follow the structure just described; the second-most common structure has a noun as root and has a verb relating to this noun as an adnominal clause (acl). For these sentences we find the acl of the spaCy dependency root and have the tree root point to the node corresponding to this acl, with an object arc pointing to the entity node corresponding to the spaCy root. All other sentence structures are not turned into referent trees. For these outlier sentences we rely on the baseline prediction method (discussed in section 2.1) instead.

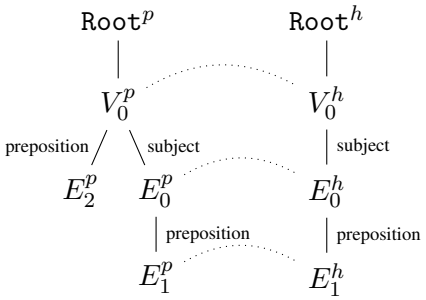
2.2.1 Comparing Referent Trees

To determine the relation between a premise and its hypothesis, we start from either tree's root and try to match nodes in the hypothesis to nodes in the premise, relying on node types and arc labels. The intuition is that for a premise to entail its hypothesis, all nodes in the hypothesis tree must be equally general or more general than their corresponding node in the premise tree. For example, 'A man walks.' entails 'A man moves.' since the matching entity nodes (both for *a man*) are equally general and the hypothesis's verb node (for *moves*) is more general than the premise's (for *walks*). Any nodes in the premise tree that have no corresponding node in the hypothesis tree simply make the premise more specific and thus are of no concern to the matter of entailment.

In practice the model recurses down both trees starting from the root nodes, matching children of the current node in the hypothesis to children of the current node in the premise based on their type (*entity* or *action*) and the label of the arc pointing to them (*object*, *subject*, or *preposition*). In this recursion the matched nodes are compared based

Node	Referent
$\texttt{Root}^p$	N/A
E_0^p	a man
E_1^p	a red life vest
E_2^p	a canoe
V_0^p	is riding

(a) Node-referent correspondence for premise 'A man in a red life vest is riding in a canoe.'.



(b) Premise tree p (left) and hypothesis tree h (right). Dotted lines indicate correspondence. Note E_2^p has no corresponding node.

Node	Referent
$\texttt{Root}^h$	N/A
E_0^h	a person
E_1^h	a life vest
V_0^h	is riding

(c) Node-referent correspondence for hypothesis 'A person in a life vest is riding.'.

Figure 1: Referent trees and their correspondence for an example premise-hypothesis pair.

on the lemmas of the tokens associated with them. Based on this comparison each matched node pair returns tallies for indications of the labels *entailment*, *contradiction*, and *neutral*. These tallies are then added together and aggregated as will be explained in section 2.3. The comparison of matched nodes happens as follows: for each lemma l_1 associated with the hypothesis node, the system aims to find a lemma l_2 associated with the premise node such that l_1 and l_2 stand in some relation indicating one of the three labels, such as co-hyponimy or synonymy. You will find what these relations are and which labels they indicate in section 2.3. If such a relation is found, a tally for the label indicated by it is incremented. Any lemma associated with the premise that has no match in the hypothesis node is counted along with the *entailment* indications, as these lemmas add specificity to the lemma that entails the same referent without said specificity (e.g. *young person* 'entails' *person*). In this comparison, auxiliary verbs and determiners are disregarded.

2.3 Combined System

The final system is a combination of the baseline system and the tree system. The tree system is used for problems in which both the premise sentence and the hypothesis sentence can be translated into trees. The baseline system is used for problems that contain a premise or hypothesis that cannot be translated into a tree. An example sentence for the baseline system is 'Two young, White males are outside near many bushes.'. The root is 'are' and its POS tag is auxiliary verb (AUX). However, AUX is not supporting the main verb, but it is the main verb itself. Since the tree system does not consider sentences containing only auxiliary verbs, the baseline system deals with these sentences.

Both systems use the same functions (i.e., relational antonyms, synonyms, hypernyms-hyponyms, see Section 2.1.1) to determine whether a word from the premise is more general than a word from the hypothesis or vice versa. To find co-hyponyms, different functions are used for the baseline and for the trees, as described in section 2.1.1. Importantly, we decided to use the same rules for cases when "no" is part of the sentence. In these sentences, the labeling should be reversed for the words the "no" is referring to. However, considering that in ten thousand examples in the SNLI training set we could only find six problems containing "no", we concluded that adding a rule for it would not be necessary.

The labeling for a problem in the tree system is as follows:

- *contradiction* if a contradiction is found in one of the nodes;
- *entailment* if there are no contradictions found in the nodes and *entailment* is the majority of the labels for the nodes;
- *neutral* if there are no contradictions found in the nodes and *neutral* is the majority of the labels for the nodes.

For example, using this labeling for the problem in Figure 1, the V_0 nodes are compared to each other, giving one entailment. Subsequently, E_0 in premise and hypothesis are compared to each other, since they both have a "subject" relation to the verb. This comparison leads to another entailment. Finally, E_1 in both premise and hypothesis are compared ("preposition" relations), giving another entailment. As no contradictions could be found and the majority label is entailment, the problem is labeled as entailment.

The labeling for a problem in the baseline system is as described in Section 2.1.

3 Results

In this section, the results from the experiment as described in the Method Section are disclosed. The baseline, tree and combined system were tested on the SNLI test set, consisting of 9824 NLI problems. Accuracy, precision, recall, and f-score for the three systems can be found in Table 4.

3.1 Baseline

As can be seen in Table 4, the accuracy when using only the baseline system on the test data is 40.9%. The confusion matrix shown in Table 1 indicates that the model performs best for entailment tasks and its performance is lowest for contradiction tasks.

Gold \ System	C	E	N
CONTRADICTION	.12	.52	.36
ENTAILMENT	.015	.77	.22
NEUTRAL	.017	.66	.32

Table 1: Confusion matrix of the baseline system on the test data

3.2 Trees

Out of the 9824 problems in the SNLI test set, 6780 problems could be converted into trees. When only looking at these problems, the accuracy of the system is 47.4% (Table 4). The confusion matrix in Table 2 provides more information on the performance for the different labels, showing that its performance is highest for contradiction tasks and lowest for entailment tasks.

Gold \ System	C	E	N
CONTRADICTION	.79	.098	.11
ENTAILMENT	.44	.41	.15
NEUTRAL	.60	.20	.20

Table 2: Confusion matrix of the tree system on the test data

3.3 Combined

When combining the tree model and the baseline model to look at the performance of the combined system, the accuracy is 44.9% (Table 4). The performance for the different labels can be found in the confusion matrix in Table 3.

Gold \ System	C	E	N
CONTRADICTION	.58	.24	.17
ENTAILMENT	.30	.52	.18
NEUTRAL	.41	.35	.24

Table 3: Confusion matrix of the combined system on the test data

System \ Metrics	Acc.	Prec.	Rec.	F-sc.
BASELINE	.409	.517	.404	.359
TREES	.474	.483	.469	.441
COMBINED	.449	.441	.447	.434

Table 4: Table of the performance of the three systems on the test set (accuracy, precision, recall, f-score)

4 Discussion

Despite the practical difficulty of using rule-based systems for Natural Language Inference, our final model achieves 44.9% accuracy, surpassing our expectations.

The first system that we built, the baseline system, is based on the spaCy dependency trees. Its accuracy is 40.9%, suggesting that it can label inference tasks above chance. This could indicate that the rules we implemented might capture some of the human reasoning applied in the SNLI dataset. However, its performance varies a lot for the different labels. While it classifies 32% of the neutral cases and 77% of the entailment cases correctly, it only finds 12% of the contradiction cases. This might be because the system only labels a problem as a contradiction when it finds a (relational) antonym or a co-hyponym. This is not often the case, since the number of antonyms in WordNet is limited, and the co-hyponym function only looks at the direct hypernyms of the words. For example, for the problem (*A man walks into a transportation station and two officers in neon jackets stand guard.; The man is at a school.*), the system predicts *'neutral'*, which indicates that it is not able to find that *'school'* and *'transportation station'* are co-hyponyms and it should therefore be a contradiction. Moreover, another limitation of the baseline system is that it only considers the root node, its children, and its grandchildren. This means that it does not scan the whole sentence, and might miss words that would contradict each other. An example of this would be the problem *(An emergency worker directs a man pulling a sled with emergency equipment on a snowy path. ; An emer-*

gency worker is at the beach.), which the system labels as entailment.

In order to improve performance and find a solution to the issues with the baseline model, we implemented the tree system. It can build trees for about two-thirds of the test set, increasing the accuracy on those problems to 47.4 %. By comparing nodes with multiple words to each other, the whole sentences are traversed, and together with an improved co-hyponyms function, the accuracy for contradictions is increased to 79%. An example of this would be: *(A little boy is getting is birthday cake and is blowing out the candles. ; A girl at school getting a ruler.)*. This example was incorrectly labeled as *'entailment'* in the baseline system, as the system does not traverse the whole sentence and cannot make the comparison between *'cake'* and *'ruler'*. With the tree system, the sentences are now traversed completely and the problem is correctly classified as *'contradiction'*. However, entailment and neutral problems are now less accurate, with 41% accuracy and 20% accuracy. This suggests that the model tends to label problems as contradictions. One possible explanation for this is that whenever a co-hyponym is found, the model labels the problem as a *'contradiction'*, even though this is not always accurate. An example for this is: *(A woman is petting a dog outside. ; A woman is rewarding a dog for bringing back a thrown stick.)*, which would be correctly labeled as *'neutral'*, but because of the co-hyponyms *'pet'* and *'reward'*, it is labeled as *'contradiction'*.

Together, the two systems achieve an accuracy of 44.9 % on the whole test set. From this, we can conclude that the rules we used to find the correct inference labels are informative and should be able to generalize to some degree. Moreover, the improvement when implementing the trees shows that comparing the nodes, which contain multiple words for each entity or verb, is beneficial in finding relations between the two sentences and determining the inference type. Lastly, while the accuracy for contradictions decreases to 58% when including the baseline system, the accuracies for entailment and neutral increase.

Despite the improvements compared to the baseline system, the final combined model of trees and the baseline has some limitations. For instance, the co-hyponyms function for the trees works well for labeling contradictions but there are cases in which the function is too strict. Both verb pairs *'sing'* and *'whisper'*, and *'pet'* and *'reward'* are labeled as *contradiction*. The first true label is indeed *contradiction* while the latter is *entailment*. In addition, spaCy tree dependency chooses only one verb when it is given a complex sentence with two main verbs. The system reaches the second verb if and only if it is a grandchild. Finally, while our system is inspired by monotonicity, it does not include many rules specific to monotonicity. Due to the limited availability of under 0.001% of the key monotonicity words "no", "every", or "some", rules for this set of words were not implemented.

5 Conclusion

This paper aimed to create a rule-based system for the Natural Language Inference task. Past research has illustrated the drawbacks of generative language models, arguing that different approaches should not be neglected (Bender et al.). Our results show that even though a rule-based system has its limitations, research on monotonicity-based systems can be useful and could lead to more insights into human reasoning.

6 Acknowledgment

The work was equally distributed between all group members (coding and writing). Data preprocessing was mainly done by Yulia, data selection and filtering of individual examples with an improvement of the code was mainly done by Shi, the word comparison functions were created mainly by Lara. Referent Trees generation was done by Jeroen. The merging of the code was done by Shi and Lara. Code polishing and commenting are mainly done by Yulia.

The report was also group work with everyone contributing to all discussions and parts of the projects, however, each of us wrote the sections relating to his/her main focus.

References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623. Association for Computing Machinery.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference.

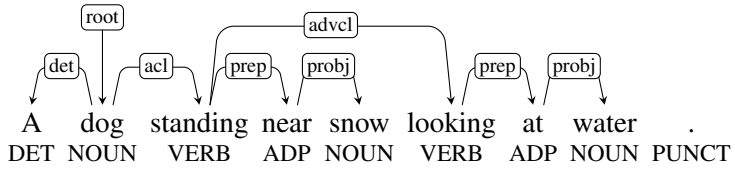
Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

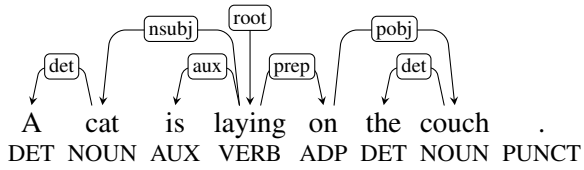
Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP.

7 Appendix

The implementation of the rules as well as the tests and statistics can be found on GitHub.

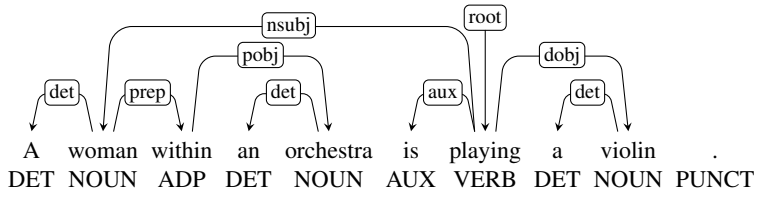


(a) Premise

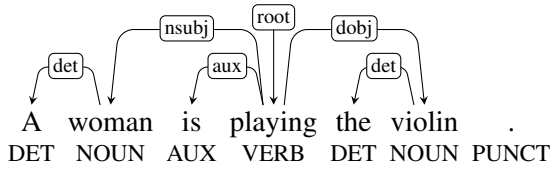


(b) Hypothesis

Figure 2: Dependency trees created with spaCy for an example from the test set of SNLI



(a) Premise



(b) Hypothesis

Figure 3: Dependency trees created with spaCy for an example from the test set of SNLI