RESEARCH MASTER THESIS

IEVA STALIŪNAITĖ

RMA Thesis: Learning about Non-Veridicality in Textual Entailment

Ieva Staliūnaitė

Supervisor: Dr. Tejaswini Deoskar
Second Reader: Dr. Rick Nouwen

Research Master Linguistics
Utrecht University
2018

Abstract

Neural network based models are the state-of-the-art for Recognizing Textual Entailment (RTE) and have recently received much attention, yet little research has addressed the question of specifically what linguistic phenomena are learned by these models. Hence, this thesis analyzes what a neural RTE model learns about items that block entailment (non-veridical operators) and whether the model can be expanded to cover this linguistic phenomena. Thus, a neural model with Long Short-Term Memory (LSTM) is trained on general natural language inference (NLI) data and tested on data from the domain of annual reports, which are written in a particular register and contain many non-veridical operators. The general domain Stanford Natural Language Inference (SNLI) data is used for training the model. An analysis of the LSTM's attention mechanism is performed in order to investigate precisely what the model pays attention to. In order to see whether the model can be improved, two datasets are added to the training set. Firstly, texts similar to the test domain are used in training, to see whether the model can learn features of the relevant register. Secondly, a dataset containing many non-veridical operators is used to train the model, to test whether the model can learn to deal with items that block entailment.

For producing the latter training set this thesis suggests a method of recasting event factuality corpora, which is abundant with non-veridical contexts. Training the RTE model on factuality data enables it to perform the task of event factuality. This thesis proposes to address both the task of specified event extraction and event factuality in one step by testing sentences about events against informative sources for entailment. The events studied in this thesis are achievements of companies with regard to the Sustainable Development Goals (SDGs).

The main contributions of this thesis are insights into the inner workings of a neural RTE model and high performance on the task of finding information about events in text. Firstly, this study shows that a textual entailment model trained on general data does not perform well on annual reports data which contains high instances of non-veridicality, and needs to be adapted. Secondly, I show that the model achieves high performance using a combination of the linguistically specialized *Veridicality* set and the domain-specific *Annual Report* datasets in training. Namely, combining these two training sets, an F_1 score of 87.05 is achieved in determining entailment between sentences in annual reports and events of accomplished SDGs. The analysis of the attention mechanism of the model shows that the model is able to induce the importance of non-veridical operators for textual entailment. Thirdly, it appears that a semi-artificially constructed recast *Veridicality* data cannot be successfully combined with the more general *SNLI* data for training a neural RTE model, on account of the recast data being too homogeneous.

Contents

Acknowledgements

First and foremost, I am very grateful to my supervisor, Tejaswini Deoskar, for interesting discussions, patient feedback and many opportunities to learn throughout the process of writing this thesis. I would also like to thank the second reader of this thesis, Rick Nouwen. Furthermore, I wish to give thanks to my supervisors and colleagues at Statistics Netherlands – Piet Daas, Ali Hürriyetoğlu, Dick Windmeijer and Marco Puts, for their trust and help during my internship.

In addition, I appreciate the privilege to have been able to learn from the amazing teachers and the clever students of the Linguistics Research Master program at Utrecht University as well as other universities that the wonderful Dutch education system allowed me to briefly attend in the past two years. I also thank the Sasakawa Young Leaders Fellowship Fund for funding my master's degree. Furthermore, I would like to acknowledge the SURFsara center, specifically the Lisa cluster and Kees van Eijden from SURFsara Credits for making it possible for me to run the experiments for this thesis.

I would also like to thank my partner Jori Jansen for the technical and design-related help as well as infinite moral support. Last but not least, I am very grateful to my family and friends, who have never doubted me.

1 Introduction

1.1 Motivation

The problem of *recognizing textual entailment* (RTE) [Dagan et al., 2013] or *natural language inference* (NLI) [Bowman et al., 2015] is an important and difficult semantic task and it is widely studied among researchers in natural language processing (NLP). Entailment is a type of inference that could be informally described as 'necessary conclusion'. The task of recognizing textual entailment is to determine whether one sentence entails another sentence, or in other words whether the second sentence (*hypothesis*) is a necessary conclusion of the first sentence (*premise*).

Humans and automatic systems alike use entailments for gaining information from natural language, however automatic systems are not yet as good at detecting them as humans and it is harder to know their underlying reasoning. Automatic textual entailment systems receive sentence pairs as input and provide as output a label saying whether there is an entailment relation between them. For instance, the sentence pair in example 1 will be labeled with the presence of an entailment relation because if something was wiggled, it follows necessarily that it was moved.

(1) PREMISE The dog *wiggled* its tail.
HYPOTHESIS The dog *moved* its tail.

For solving this task, most NLP systems of textual entailment use machine learning methods with large amounts of sentence pairs annotated for the presence or absence of entailment. Recent work in RTE research mostly focuses on neural models and has largely been directed to improving performance rather than studying the linguistic features of the sentences. Neural models are data-driven, which means that they use very large sets of sentence pairs and abstract relevant features from the input by using the labels as a reference point. An advantage of neural RTE systems in comparison to logical or feature-based ones is that they yield very high results. On the other hand, there is little knowledge about what linguistic representations they build and hence what drives the high performance of the neural models.

In example 1, for instance, a human classifying the sentence pair with regard to an entailment relation would rely on the subset-set relation between the words 'wiggle' and 'move'. Research has shown that neural textual entailment models also rely on such taxonomic relations [Rocktäschel et al., 2015]. However, in many other cases it is not known what neural models depend on when classifying sentence pairs. Even though the problem of textual inference has been widely studied for over a decade, the question of what linguistic information neural NLI systems rely on has only started receiving attention very recently [Williams et al., 2017, Lai et al., 2017, Poliak et al., 2018a, Poliak et al., 2018b].

Long Short-Term Memory (LSTM) units are used in the architecture of some neural models and are particularly interesting when analyzing what linguistic

information the models are able to capture. LSTMs have been specifically developed to represent sequential data in neural models [Hochreiter and Schmidhuber, 1997]. They have been very successful at improving various types of NLP models and they were recently shown to also work well in various semantic tasks including textual entailment [Palangi et al., 2014, Le and Zuidema, 2015, Chen et al., 2016]. LSTMs can capture long-distance relations between items in a sequence, such as words, characters or part of speech tags, by encoding every item while taking into account its context. In addition, *attention mechanisms* can encode which context information is important for each item for the given task. LSTMs with an attention mechanism are used in the RTE model in this thesis and facilitate the analysis of the semantic phenomena that the model captures. As they are able to capture relations between words, this thesis looks into not only lexical semantic relations but also structural relations such as subordination, negation and modification, in relation to entailment.

This thesis makes use of the advantages of neural textual entailment systems as well as attempts to contribute to eliminating their drawbacks. I use the advantage of the high performance of textual entailment systems in adopting the RTE method to solving other tasks such as specified event extraction and event factuality, which are less studied. In addition, I try to shed light on the black box that neural models are said to be by analyzing whether the model can correctly classify examples with a specific linguistic phenomenon as well as what semantic relations the model deems important in these decisions. These questions are worthwhile to investigate in order to improve RTE models as well as learn about their scalability to different test sets.

1.2 Main Concepts

This thesis focuses on the phenomenon of *veridicality*, which is one of the factors that have an effect on the *entailment* relation between two sentences. Defining what exactly it means for one sentence to *entail* another sets the gold standard – the guideline for how sentence pairs should be labeled with regard to entailment. This step is important for annotating the data for training and testing an RTE model. On the other hand, defining *non-veridical* operators allows one to determine whether the presence of these operators causes difficulty for an RTE model and whether the model attends to them in classification. In this section I first introduce the definition of entailment and its interpretation in NLP. Then I present the concept of non-veridicality and discuss the scarcity of this concept in the research on textual entailment so far.

In NLP tasks the notion of textual entailment is used to denote the inference based on the *strong likelihood* that one sentence is true if another sentence is true, as described in the RTE challenges [Dagan et al., 2006]. In contrast, according to the logical definition, the entailment relation between two sentences is more restricted, ensuring that the hypothesis follows from the premise *necessarily*:

Definition 1 *Sentence* A *entails sentence* B *if and only if whenever* A *is true,* B *must also be true.*

While the loose NLP definition of textual entailment is sufficient for many practical applications, in some cases more precise inference is necessary, for example in the domain of science or legal issues. This research is carried out in the domain of *annual company reports*, therefore I follow the logical definition when evaluating entailment between two sentences. For instance, in example 2 the premise does not entail the hypothesis in the logical sense, as claims are not necessarily true.

(2) PREMISE Researchers *claim* to have transferred a memory from [1] one sea snail to another.

HYPOTHESIS Researchers transferred a memory from one sea snail to another.

This example pair illustrates the effect of non-veridical operators as the word 'claim' blocks entailment between the sentences. Non-veridical operators are the operators that do not fall into the veridical class according to Definition 2 paraphrased from [Giannakidou, 1999]:

Definition 2 *Veridicality is the quality of a sentential operator* O*, such that* Op *entails* p*, where* p *is a predicate.*

Hence, the verb 'claim' (O) in the premise in example 2 is a non-veridical operator, because 'transferred a memory from one sea snail to another' (p) is not entailed by 'claim to have transferred a memory from one sea snail to another' (Op). Various types of non-veridical operators have been studied extensively by semanticists [Valencia et al., 1993, Giannakidou, 1999, Giannakidou, 2002, Karttunen and Zaenen, 2005, Giannakidou, 2006, Penka and Zeijlstra, 2010, Zaenen and Karttunen, 2013], and include adverbs ('hardly', 'supposedly', etc.), negation ('never', 'not', etc.), propositional attitudes ('think', 'doubt', etc.), modals ('will', 'should', etc.), connectives ('or', 'if', 'before', etc.), questions, etc.

Most neural RTE systems are trained on data that does not encode the relevance of non-veridical operators. For instance, SNLI (the Stanford Natural Language Inference corpus [Bowman et al., 2015]), the largest NLI dataset available, consists of more than 500 000 sentence pairs, yet none of them encode the effects of the non-veridical operator 'can'. While the modal 'can' is present in some sentences in the SNLI corpus, it never appears in the position where its presence is important for the entailment relation between the two sentences. For instance, the sentence pair from SNLI in example 3 has the mention of the word 'can' in the premise, yet the relation between the premise and the hypothesis does not depend on the presence of the modal.

(3) PREMISE A man in a suit is painting a picture while outside, a clock *can* be seen in the background above the hedges.

[1] Adapted from https://www.theguardian.com/science/2018/may/14/scientists-transplant-memories-between-sea-snails-via-injection

HYPOTHESIS The man is painting a picture for his client.
[Bowman et al., 2015]

In contrast, in example 4 the modal 'can' determines the absence of entailment between the sentences. It is interesting to see whether a model that is trained on SNLI data would be able to make inferences about sentence pairs in which non-veridicality is a deciding factor of entailment.

(4) PREMISE I *can* fly.
HYPOTHESIS I fly.

1.3 Research Questions and Contributions

The research questions of this study are as follows:

(5) RQ1 *What does a neural textual entailment model learn about specific linguistic aspects such as veridicality?*
RQ2 *Can such a model be directed to learn the features of language that are relevant to a particular domain by adding specialized training data?*

In answering these questions, this thesis provides insights into the coverage of neural textual entailment models with regard to linguistic phenomena. To answer RQ1, I test how the neural RTE model trained on general NLI data performs on a specialized set of annual reports of companies, which are written in a particular register and contain many non-veridical operators. The analysis of the results and the attention mechanism provides some answers to the question of what the neural model learns about language.

In order to answer RQ2, this thesis tests whether the performance of the model can be improved by including specialized datasets in the training of the RTE model. A *domain-specific* dataset of annual reports and a *linguistically specialized* data is tested with a dataset comprised of many non-veridical contexts. The contribution of adding the datasets to the training set of the model is evaluated by analyzing the results of the model and the outputs of the attention mechanism.

Annotated data capturing the effects of non-veridical operators is not available in a format suitable for training RTE systems. One contribution of the thesis is proposing a method for recasting event factuality data to textual entailment-style labeled sentence pairs. Factuality corpora encodes the factuality of events described in sentences and is usually published in the format of example 6. This data contains many non-veridical operators, such as 'said', 'would' and 'allow' in the sentence in example 6, which inform the reader that the event of leaving is hypothetical.

(6) SENTENCE Soviet officials also *said* Soviet women, children and invalids *would* be *allowed* to leave Iraq.
EVENT WORD leave

LABEL non-factual
[Saurí and Pustejovsky, 2009]

In order to transform this data into premise-hypothesis pairs, I use dependency parsing and devise an algorithm to find the span of the sentence that refers to the given event. As a result, the hypothesis in 7 is produced to denote the leaving event in 6.

(7) PREMISE Soviet officials also said Soviet women, children and invalids would be allowed to leave Iraq.

HYPOTHESIS Soviet women, children and invalids leave Iraq.

The algorithm for the generation of the *Veridicality* dataset by converting factuality examples is described in Chapter 4.

1.4 Using the RTE Method for Detecting Factual Events

In this research, the methodology of RTE is used for finding information about the factuality of specific events in text. This task has traditionally been addressed by a two step approach – first *extracting* the relevant event, and then determining whether it is *factual* or not. By using the recast factuality data described above and adopting the RTE method, some of the drawbacks of the conventional approach could be averted, as shown below.

If one wanted to find the event of 'memory transfers between two animals' in the text in example 8 with the conventional specified event extraction method, they would have to define the words that could denote the participants of the event and determine whether the relation between these entities is that of 'transferring'.

(8) Researchers *claim* to have transferred a memory from one sea snail to another.

One drawback of this method is that the participants of the event could be described in many different ways which would all have to be spelled out in order to find the event. Specified event extraction has been performed extensively in the biomedical domain in which names of entities are rather normalized and the events are quite specific [Kim et al., 2009, Kim et al., 2011, Nédellec et al., 2013]. However, in other domains it can be more difficult to define all the ways that an event might be described in. For instance, in the case of the example above one would have to search for all the hyponyms of 'animal' in the text.

Another drawback of the approach of event extraction is that it requires another step of making a decision about event factuality after the event has been extracted. For example, the event of a memory transfer in sentence 8 is possible yet not factual as it is just a claim. It would be more efficient to have a system that finds the relevant event in the text and determines whether it is factual, all in one step.

When using a textual entailment system for this task, both the issue of defining event participant forms and the necessity for a two-step system can be avoided. More specifically, the event could be described as a hypothesis and tested against the premise sentences in the text to determine whether the texts entail the occurrence of the event. For instance, if the event of interest is memory transfers between animals, one can formulate the hypothesis in example 9 and test it against information sources such as the premise sentence.

(9) PREMISE Researchers claim to have transferred a memory from one sea snail to another.

HYPOTHESIS Memories were transferred between two animals.

A textual entailment system is suitable for finding the event in this case as it is able to detect the taxonomic relation between 'animal' and 'snail' and thus can conclude that the event of memory transfers between snails is an instance of memory transfers between animals. Furthermore, if the textual entailment system is able to recognize non-veridical contexts, it can also determine that this event is not factual, by taking into account the non-veridical operator 'claim'. The system is effectively finding the event and evaluating its factuality by determining that there is no entailment in example 9.

1.5 Domain

In this thesis the method described in the previous section is used in the domain of *annual reports* (AR) of companies. The practical goal of this thesis is to extract information from annual reports of companies about their performance with regard to the Sustainable Development Goals [SDGs, 2015] set out by the United Nations. Specifically, this research focuses on *Goal 13: Climate Action*. The targets include reducing emissions, water and energy consumption, and waste production. The accomplishments of the companies with regard to these targets are treated as events to be extracted. Extracting these events and determining their factuality could help to automatically determine to what extent the companies have achieved the targets of the SDGs.

A model of textual entailment is a suitable substitute for the event extraction and factuality methods in this particular domain because of the pervasiveness of non-factual events in the Annual Report data. The non-factual events described in the Annual Report data are frequently embedded under non-veridical contexts such as 'aim', 'potential' and 'will'. For instance, sentence 10 presents the potential achievements of the company rather than actual ones.

(10) An energy efficiency program in Italy has *potential* energy savings of 1.5% of total energy consumption before the end of 2015 - equal to an 8.000 ton reduction of CO2 emissions. [Refresco Annual Report, 2014]

It would be preferable to only extract the events that refer to actual achievements and not only aims and dedication statements. The distinction between

the results that have been achieved and mere intentions or attempts is important in this task for the evaluation of the performance of the companies.

This domain is difficult for the following two reasons. Firstly, the texts are written in a special register, which means that they contain vocabulary and structures that are rare in general NLI data. This could be expected to cause difficulty for a general RTE system trained on SNLI, for instance, as there would be many unseen expressions. Secondly, the difference between strong implicatures and entailment is very subtle in this domain. For instance, a company claim is presented in the premise in example 11, suggesting positive changes.

(11) PREMISE The *steps we have taken* to reduce our environmental footprint strengthen our credibility as a responsible company. [KPN Annual Report, 2015]

HYPOTHESIS Our environmental footprint was reduced.

However, the hypothesis in example 11 is not entailed because steps taken towards some goal are not sufficient to conclude that the goal has actually been achieved. If all the sentences from the annual report are tested against this hypothesis, a general conclusion about the factuality of the event can be made. For instance, if none of the sentences in the report entail this hypothesis, the company must not have reported any reductions of their environmental footprint.

1.6 Model and Experiments

I adopt the model described in [Rocktäschel et al., 2015] for recognizing entailment between sentences, namely a recurrent neural network model with Long Short-Term Memory (LSTM) cells and a word-by-word attention mechanism. This model is chosen on account of having achieved the state-of-the-art results in the task of textual entailment recognition. Aside from high performance, the model with an attention mechanism enables inquiry into the inner workings of the neural model.

In Experiment 1 (Chapter 5) I test how this textual entailment model performs on the specific Annual Reports data, in order to answer the first research question (*what does a neural textual entailment model learn about specific linguistic aspects such as veridicality?*). I analyze the attention scores in order to see what the model finds important in making textual inference decisions in an unknown domain. In the remaining 3 experiments, I use a combination of various corpora in training the model with a view to answering the second research question (*can such a model be directed to learn the features of language that are relevant to a particular domain by adding specialized training data?*). I compare what the model learns when it receives training data that covers certain phenomena, such as domain-specific Annual Reports data in Experiment 2 (Chapter 6), and Veridicality data in Experiments 3 and 4 (Chapters 7 and 8, respectively).

2 Discussion of Related Work

This chapter describes the methods, results, advantages and drawbacks of the relevant computational work. Firstly, I review the *textual entailment* systems that have been built so far, weighing the advantages of neural vs. logical and feature-based systems, with a view to finding a future direction for such models. Secondly, I discuss the studies in the domain of *event factuality* and the corpora that have been built for this task, which could be adapted to the current study. Thirdly, I give a brief overview of the application area of *fact checking* that relates to this study, since it involves the task of event factuality and could potentially benefit from being addressed with an RTE system. Finally, I present the few recent studies that have addressed the question of *what neural textual entailment models can learn about semantics*, and discuss the further steps for developing this line of research.

2.1 Textual Entailment

First off, some studies rely on formal semantics in order to detect textual entailment by searching for precise logical relations between sentences. Such studies do not treat highly likely or implied statements as entailed, which is a virtue that this thesis also aims to follow. [Bjerva et al., 2014] reaches the highest accuracy of logical systems in this task, namely 82% in classifying sentence pairs with regard to whether or not the premise entails the hypothesis. They convert the texts to a first-order logic representation and search for contradictions or entailment relations using theorem provers. This approach is able to deal with non-veridical operators as they determine the logical relations in the sentence. The main drawback of this this research is that converting various linguistic structures to uniform logical form is labour-intensive as well as not cross-linguistically transferable.

In contrast, other studies use hand-crafted features and apply machine learning algorithms to detect textual entailment, thus relying on probability more than exact relations. For instance, [Zhao et al., 2014] use a feature-based model that estimates similarity between words and sentences in order to detect textual entailment. They lemmatize all words and replace all synonyms by the same base word. Zhao et al. use diverse features such as presence of antonyms or negation, longest overlapping strings, a sequence of relation units from dependency trees, etc. They achieve an accuracy of 83.46% in recognizing entailment relations, which is an improvement on the formal logic method. However, this method is also not very efficient as it requires designing the features and computing their values before classification can be performed.

The current textual entailment research has moved towards even more automated methods, approaching the task with neural models. For instance [Rocktäschel et al., 2015] use Recurrent neural networks (RNNs) with long short-term memory (LSTM) units and a word-by-word neural attention mechanism. This model performs at an accuracy of 83.5% in recognizing entailment. Such a model is not explicitly provided with the logical structures or semantic features,

and instead receives a huge amount of labeled sentence pairs and learns to determine entailment relations between new sentences by abstraction. Overall, neural models reach better performance than formal logic approaches or the hand crafted feature systems. In addition, Rocktäschel et al. analyze the attention mechanism of the model and show that the model is able to learn lexical semantic relations such as synonymy and hyponymy. However, the question of whether the model is able to learn other semantic relations, such as veridicality, remains unexamined.

In addition, some research has applied the method of RTE to other tasks in natural language processing. For instance, [Harabagiu and Hickl, 2006] investigate whether the textual entailment method could improve the performance of question answering systems. The study follows the definition of question answering in Definition 3 taken from [Groenendijk, 1999]. They show that indeed by testing whether potential answers are entailed by the given question, question answering systems can be improved by 20%.

Definition 3 p *is considered to be an answer to a question* ?q iff ?q *logically entails the set of worlds in which* p *is true*

By and large, textual entailment systems play a role in other NLP tasks that involve semantics, yielding good results even with feature-based systems. It is therefore interesting to investigate how neural textual entailment systems can be applied to other tasks too. Moreover, even though neural systems outperform the more traditional logical approaches, not much is known about the inner workings of these models. This means that the models are susceptible to the menace of overfitting – the potential dependence on some idiosyncratic and/or trivial features of the given data by the model, without actually learning the semantic relations between sentences.

2.2 Event Modality

The task of finding out whether an event mentioned in a text has happened, could have happened or did not happen is referred to as *event modality* or *event factuality*. It is very useful for reducing human effort in finding information about questionable events. For example, an event factuality system could determine that the event denoted by the word 'won' in sentence 12 is non-factual as people's thoughts are not always reliable in accurately reflecting reality.

(12) They *thought* that she won the Nobel Prize.

Numerous studies use the concept of non-veridical operators to find out which contexts change the modality of an event (e.g. the verb 'thought' in example 12). The term *hedge* is sometimes used instead of *non-veridical operators*.

Some studies address the task of determining which words are hedges by building rule-based systems or defining features that could indicate the non-veridicality of words [Hutchinson, 2004, Özgür and Radev, 2009]. These studies use features such as lexical form, POS tag, subordinating predicate class, etc.,

and reach F-scores in the range of 82-91% in classifying words with regard to whether or not they are non-veridical. This approach yields fairly good results, however the task of event factuality is not yet solved as this method only tells the researchers whether some words are hedges but not whether the event that is hedged is factual. For instance, the non-veridicality of the word 'fail' does not determine whether the event denoted by 'enter' in example 13 is factual.

(13) She did *not fail* to enter.

In fact, the entering event can be deduced to be factual from the sentence even though 'fail' is non-veridical, given that the hedging effect is cancelled by another non-veridical operator 'not'. Özgür et al. also tackle the task of detecting the span of the hedge, however the achieved results are not high. Following this approach for event factuality one would additionally have to find which events the discovered hedges scope over and determine how multiple hedges interact with each other.

In contrast, [de Marneffe et al., 2011] investigate event modality directly instead of depending on the operators being classed as non-veridical. They reach an 83% micro-averaged F1 score in event factuality classification. De Marneffe et al. use linguistic features such as the lemmas of words, negation, modality, conditionals, etc. and examine the path from the event word to the root in the dependency parse for their analysis. In addition to it, they include the lemma of the subject as a feature in classifying event factuality, in order to account for world knowledge, as people find some sources more reliable than others, as illustrated by examples 14a and 14b.

(14) a. The FBI said it received . . .
b. Bush said he received . . .
[de Marneffe et al., 2011]

In this thesis, however, the aspect of the trust-worthiness of the source of a claim is not relevant, because events in a report are defined more strictly and no source can be taken to be more or less questionable than another. Therefore, this thesis focuses only on the linguistic features of the sentences themselves, excluding world knowledge.

Instead of treating factuality as a binary feature as in the previously discussed studies, [Lee et al., 2015] use a scale to represent the level of factuality. The dataset used in this study contains manual annotations of non-experts with regard to whether or not a particular event mentioned in text has happened. The annotators find the events mentioned in text and provide a score from -3 to 3 based on how certain they are that the event has happened. The researchers then use lemmas of the target words, parts of speech, hyponyms, Brown clusters and dependency paths to predict the factuality of a given event, reaching an F_1 score of 70.8. While the score is lower than that of [de Marneffe et al., 2011], this study solves a harder task, producing a more fine-grained distinction in event factuality levels. However, this is not desirable for this thesis, wherein entailment is defined in logical terms. Lee et al. interpret event factuality in

the same way as textual entailment has been dealt with in NLP applications, treating high correlation as not qualitatively differ different from fact.

Neural models appear to outperform the feature-based ones in this field as well. [Rudinger et al., 2018] use Neural Models (a stacked bidirectional linear chain LSTM and a stacked bidirectional child-sum dependency tree LSTM) to predict the factuality of events. They achieve state-of-the-art results, with a Pearson correlation (r) score of 0.857 in classifying events as factual or not. Rudinger et al. show that the linear chain-structured network outperforms the tree-structured network and conclude that the syntactic tree information is not necessary in determining the factuality of events. This approach is the most cost-effective as features do not have to be devised and factuality is evaluated directly, without relying on the classification of hedges. Moreover, this research shows that some features that appear important intuitively, such as the syntactic structure, might not actually be beneficial for the system. This illustrates the benefits of neural networks, as they are able to select features that yield good results, which might not be the same ones as humans would design.

Many of the studies on event factuality provide good resources for the current thesis. For instance, [Lee et al., 2015] build the *UW* corpus based on the scaled judgments of their participants. In order to use it for this thesis, the factuality judgments would have to be made stricter to rule out high likelihood as not factual. Furthermore, the *FactBank* corpus [Saurí and Pustejovsky, 2009] contains descriptions of events labeled with certainty, probability or possibility of their factuality. Similarly, the *It Happened* corpus [White et al., 2016] provides a binary label with an indication of confidence for the factuality of events.

2.3 Fact Checking

Fact checking is an application which combines the previously discussed tasks of event modality and event extraction. While event extraction focuses on the span of text that describes the event itself, event modality research focuses on the remaining span of the sentence, searching for hedges that might make the event non-factual. In the task of fact checking one has to both find the relevant facts and then validate them. However, fact checking research has not been extensively explored yet.

Early implementations of fact checking systems simply rely on a collection of given facts. Such studies compare potential facts to a given database of true and false facts and measure the semantic similarity between them [Vlachos and Riedel, 2014, Ciampaglia et al., 2015]. This approach is not scalable because new claims cannot be tested for factuality, as a database is limited to claims that have already been evaluated. In addition to it, [Vlachos and Riedel, 2014] note that another drawback of such an approach is that the output of the system lacks grounding. That is, the reasons for a system to output a label 'true' or 'untrue' for a given fact are unknown.

Instead, I propose to use the method of textual entailment for tasks similar to fact checking, in which case only an information source text and a sentence of interest are necessary. In the textual entailment method the event of interest

is formulated as a hypothesis and the sentences in a source text are treated as the premises. This way I attempt to solve the problems that are notable in the research discussed in this chapter. The main drawbacks to be avoided are resource and labor intensity, dependence on specific resources, multiple step pipelines and partial coverage of the larger issue of finding factual events in text.

2.4 Analyzing What Neural NLI Models Learn

A number of very recent studies have probed the semantic knowledge of neural network models [Poliak et al., 2018a, Poliak et al., 2018b, Marvin and Koehn, 2018]. These studies show that neural machine translation and natural language inference systems have some knowledge of semantic phenomena such as event factuality and non-veridicality, among others. Some of these studies use the method of recasting existing corpora into sentence pair format for finding out whether a textual entailment model has knowledge of factuality and non-veridicality. For instance, event factuality items such as the sentence "I'll not say anything" with the label 'non-factual' for the event 'say' are recast into the sentence pair in example 15 with the label 'no entailment'.

(15) PREMISE I'll not say anything.
HYPOTHESIS The saying happened.
[Poliak et al., 2018b]

[Poliak et al., 2018b] use this method to build a Diverse NLI Collection (DNC) that contains various semantic phenomena. They then test a model trained on general NLI data (with or without pre-training on the DNC) on the different test sets within DNC. As a baseline for their experiments, Poliak et al. use the classification results of a model trained on only the hypothesis and discarding the premise, because recent work has shown that this is a strong baseline due to biases in the NLI datasets [Poliak et al., 2018c]. The research by Poliak et al. shows that models trained on general NLI datasets do learn something about factuality and veridicality, and in the case of event factuality can also yield even better results if the model is pre-trained on DNC. On the other hand, pre-training on the Multi-NLI corpus [Williams et al., 2017] that is diverse with respect to genres represented does not improve the performance in these tasks.

Similarly, [Glockner et al., 2018] build a test set that encodes lexical semantic relations between words and test how various models trained on SNLI and Multi-NLI perform on it. In their test set, the premises are taken from SNLI and the hypotheses are identical to the premises except for one word, which is substituted with its hypernym, co-hyponym or other related word. Glockner et al. show that when models are trained on SNLI they perform quite poorly on this new dataset. The results of this study show that the models trained on SNLI might not be learning lexical semantic information even if they are performing well on SNLI test data. This effect is additional evidence that the SNLI dataset is biased. In addition, models trained on a combination of Multi-NLI

and SNLI perform better than those that are trained only on SNLI, which shows that additional, more varied data helps neural RTE models scale with regard to the types of linguistic features they are able to learn.

This thesis addresses a similar question, namely what general NLI systems know about factuality and veridicality. However, there are some differences that set this thesis apart from [Poliak et al., 2018b]. Firstly, the neural model I use in this study encodes the premise and the hypothesis to a single representation whereas Poliak et al. encode the sentences separately and later concatenate them. The encoding of the two sentences together allows the model to represent the relations between the words in the two sentences. These relations can later be analyzed by using the attention mechanism of the model. Secondly, my recasting method yields more naturally phrased hypotheses than the one in example 15 (see Chapter 4 for detailed description). Hence, I also use this data for training the RTE model and not only for pre-training and testing.

3 Model

3.1 Architecture

This study adopts a model by [Rocktäschel et al., 2015], that was constructed for learning textual entailment and trained and tested on the SNLI [Bowman et al., 2015] dataset. The system is a neural model that takes a sentence pair as input and assigns it to the 'entailment', 'neutral' or 'contradiction' class based on the relation between the premise and the hypothesis.

The sentences are represented as sequences of word embeddings from Word2Vec [Mikolov et al., 2013], with a 300-dimensional vector encoding the meaning of each word. I use pre-trained vectors that have been generated with unsupervised techniques to represent the relations between words based on their distribution in texts [Mikolov et al., 2013]. The RTE model represents words unseen at inference time with random vectors.

In addition, the model has long short-term memory (LSTM) units [Hochreiter and Schmidhuber, 1997]. More specifically, the premise and the hypothesis are encoded into a single representation of the sentence pair using the LSTM. Namely, the hypothesis is encoded by conditioning it on the representation of the premise. This way the relations between the words in the two sentences are encoded in the sentence pair representation. Such conditioning yields better results than a setting wherein the sentences are represented separately.

Moreover, [Rocktäschel et al., 2015] use a word-by-word attention mechanism in order to encourage the model to use relations between individual words and phrases in determining entailment relations. The word-by-word attention mechanism assigns high scores to relations between words in separate sentences which are important for the classification decision. In the word-by-word mode of the attention mechanism, the model attends over the output of the LSTM for the premise while another LSTM processes each word in the hypothesis for each sentence pair. This results in a sequence of attention weights distributed across the words in the premise for each word in the hypothesis. The weights of the words in the premise are gradient-log-normalized with the softmax function. The attention-weighted representation of the premise at the time of processing the last word in the hypothesis is combined with the last output vector of the LSTM to produce the final representation of the sentence pair. Such a setting allows the model to learn relations such as synonymy, antonymy and hyponymy between the words in the premise and those in the hypothesis. In addition, the word-by-word mode of the attention mechanism yields the highest performance of the model.

The model is trained with cross-entropy loss. The hidden layers of the model are composed of 100 units. The model is tuned by performing a grid search of the best hyperparameters from the combinations of the values of initial learning rate [1E-4, 3E-4, 1E-3], dropout rate [0.0, 0.1, 0.2] and l_2 regularization strength [0.0, 1E-4, 3E-4, 1E-3].

3.2 Implementation

I use the model implemented by [Junfeng, 2016]. This implementation is chosen because it reaches an accuracy that is only 0.21 percentage point lower than the accuracy reported by the authors of the model.

I reproduce the results of [Junfeng, 2016], achieving an accuracy of 83.29% with the hidden size 100, initial learning rate 0.001, dropout rate 0.2 and l_2 regularization weight 0.0. For the experiments in this thesis, the tuning of the hyperparameters is carried out by a grid search of the best parameter setting, following [Rocktäschel et al., 2015]. Every 20 epochs the learning rate is reduced by 0.00005. The model yielding the highest performance on the development set for each training set is evaluated on the test set.

Since the implementation does not contain an interface for analyzing the attention scores of the model, I build a Python object that saves the attention scores that can be retrieved at prediction time. This object is used for the analysis of what phenomena the models attend to in the experiments carried out in this thesis.

3.3 Annotation

The number of prediction classes in this thesis differs from the model in [Rocktäschel et al., 2015]. 2 classes ('entailment', 'no entailment') are used instead of the original 3 classes ('entailment', 'neutral', 'contradiction'). The 'neutral' and 'contradiction' classes are merged together for the following reasons.

Firstly, different corpora which are used for training models in this thesis are not annotated in a consistent manner. As previously discussed, some corpora contain binary classes (with or without confidence levels) while others evaluate items on a scale. This means that in order to merge the corpora, *ad hoc* decisions have to be made about which classes correspond to which others from the different annotation types.[2]

Secondly, for tackling the tasks of event extraction and factuality, the difference between 'neutral' and 'contradiction' classes is not essential. That is, if a statement in a text contradicts some event, one cannot conclude that the event never happened. In the case of this thesis, it is possible that some company increased its water consumption at some point (for instance due to droughts in the summer), but that does not deem it impossible that the company reduced their water consumption overall. Such nuanced relations are considered in more detail in the discussion of the limitations of the study and future directions in Chapter 9.

A test set and a training set composed of sentences from Annual Reports (AR) of Dutch companies is hand-annotated by the author of this study. The annotations follow the formal definition of entailment. The annotation scheme is as follows: After reading a sentence pair, decide whether there is a possible

[2] Given more time and resources end-to-end learning could be used instead. That is, various class divisions could be used in training the model and the best performing division one could be adopted.

situation in which the premise is true and the hypothesis is not true. If such a situation is possible, the sentence pair is annotated with 'no entailment'. Otherwise, the sentence pair is labeled with 'entailment'.

For example, the premise in example 16 can be true while its hypothesis is false, because an award for leadership and efforts does not entail actual achievements.

(16) PREMISE Also in November, TNT Services UK & Ireland *won the 'Environment Award'* at the 2014 Global Freight Awards ceremony organized by Lloyd's Loading List.com, for *demonstrating leadership in efforts* to reduce energy consumption, emissions, noise pollution and environmental impact. [TNT Annual Report, 2014]

HYOPTHESIS TNT Services UK & Ireland reduced energy consumption, emissions, noise pollution and environmental impact.

Hence, the relation in example 16 is 'no entailment'. This conclusion can be verified by the fact that concatenating the premise and a negation of the hypothesis does not lead to a contradiction:

(17) Also in November, TNT Services UK & Ireland won the 'Environment Award' at the 2014 Global Freight Awards ceremony organized by Lloyd's Loading List.com, for demonstrating leadership in efforts to reduce energy consumption, emissions, noise pollution and environmental impact *even though* TNT Services UK & Ireland did not reduce energy consumption, emissions, noise pollution or environmental impact. *It was a consolation prize.*

3.4 Experiments

In this section I introduce the experiments that are conducted in this study using the model described in this chapter. Besides replicating the results of [Rocktäschel et al., 2015], I use the model with the adjustments laid out above, for training on various datasets in 4 different experiments, tuning its hyperparameters, testing on the annual reports data and analyzing the results and the attention mechanism scores.

Experiment 1 tests what the model learns if it is trained on the SNLI data. More precisely, I test the model on sentence pairs from Annual Reports in order to investigate how the model performs on a set that has an abundance of non-veridical operators and is written in a register of a specific domain. In *Experiment 2* I use Annual Reports (AR) items in training as well as testing. The items are added to the training in Experiment 2 with the view to investigating how training on domain specific data affects the performance of the model. In *Experiment 3* I use recast data containing many non-veridical contexts in training the model, in order to see whether the model can be directed to learn

a specific linguistic phenomenon. In *Experiment 4* I test whether modifying the data used in Experiment 3 to be less linguistically restricted improves the results.

3.5 Data

In this section I provide an overview of the training and test sets for the 4 experiments carried out in this research. I describe the sources of the data, the preprocessing, the sentence extraction and labelling procedures.

3.5.1 Test Data for All Experiments

The *AR* test and development sets for all experiments in this study are composed of the annual reports of a number of organizations in the Netherlands in 2017. The annual reports contain information about a company's activity and are publicly accessible. In what follows I describe the process of generating sentence pairs and their labels for these datasets.

3.5.1.1 Premise Preprocessing and Extraction

The PDF documents of annual reports are web crawled and converted to text format, and the text is sentence-tokenized with nltk [Bird et al., 2009]. Some additional processing is done to remove sentences that are longer than 82 words, sentences that are written in all capitals as well as the ones that start with a lower case letter or do not end with a sentential punctuation are excluded as likely titles of sections or incorrectly sentence-tokenized text snippets.

The premises for the test set sentence pairs are extracted from the annual reports. A total of 1742 premises are extracted from the reports of 58 companies in 2017. The sentences from the reports are selected on the basis of containing any grammatical form of any words in the set of [CO_2, water, emissions, footprint, pollution, carbon, consumption, waste, discharge, emanation, contamination].[3] In order to avoid biases based on the name of the company, the company names are substituted with 'we' in the extracted sentences.[4] For instance, the sentence 18a becomes 18b.

[3]Lemmatization is performed with *CoreNLP* [Manning et al., 2014] for finding diverse forms of the same word. The words are selected on the basis of the indicators for the Climate Action goal of the Sustainable Development Goals (SDGs). For other domains a similar list could be assembled.

[4]For substituting the names of the companies with the first person pronoun, I find named entities in the reviews (extracted with nltk [Bird et al., 2009]) that match the company names by 90%. Levenshtein distance is used (fuzzywuzzy package, https://github.com/seatgeek/fuzzywuzzy). Since the model does not have character-level representations, the lack of agreement between the verb and the subject in the resulting sentence is ignored in this thesis under the assumption that the model will be minimally affected by it. Nonetheless, some negative effect could be present as different forms of the same lemma could have slightly different word embedding representations.

(18) a. *Shell* has taken steps to improve water recycling in one area of the Permian shale asset in west Texas, USA. [Shell Annual Report, 2017]

b. *we* has taken steps to improve water recycling in one area of the Permian shale asset in west Texas, USA.

614 of the premise sentences contain potentially non-veridical contexts. That means that there is a non-veridical word or a structure that scopes over some part of the premise in about one third of the test items, not evaluating whether this has an effect on the label of the respective sentence pair.

For tuning the models, an additional set of 858 of development items from the annual reports is extracted and preprocessed following the same procedure.

3.5.1.2 Producing Hypotheses for Textual Entailment

In order to test whether the annual reports entail the events of accomplishing Sustainable Development Goals (SDGs), the hypotheses representing the achievements are generated. With the sentences from the annual reports serving as premises, hypotheses are formulated based on the aspects of SDGs that each premise mentions. The same hypothesis phrase is used for all sentence pairs, only changing the aspect word in "*Aspect* is reduced".[5] For instance, for the premise in example 19, the hypothesis about emission reduction is produced.

(19) PREMISE Smart homes are *helping us* lower our energy requirements, saving costs and helping cut emissions. [ASM Annual Report, 2016]
HYPOTHESIS *Emissions* are reduced.

The sentences are hand-annotated following the annotation scheme in Section 3.3. In the last example the label is 'no entailment', because it is possible that one entity is helping another entity to do something, yet the goal is still not achieved by them.

3.5.2 Training Data: Experiment 1

The Stanford Natural Language Inference (SNLI) [Bowman et al., 2015] corpus is used for training the textual entailment model in Experiment 1. This dataset is composed of human-written and manually labeled sentence pairs that are based on the premises extracted from social media. The SNLI corpus is composed of pairs of sentences and a label for each pair, which marks whether the premise sentence entails the hypothesis sentence. For this thesis, the labels are converted to the 2 class distinction. For example, the premise in example 20 does not entail its hypothesis, in fact the two sentences are completely unrelated.

[5] This formulation excludes the agent of the reduction event from the hypothesis. Since all the premises come from company reports, the agent of the events is assumed to be the company itself.

(20) PREMISE One man wearing a blue shirt, white shorts, and sandals, and another man wearing cargo pants and a gray jacket shop at a farmer's market.
HYPOTHESIS The brains are attacking, everyone run for your lives!

A subset of the *SNLI* dataset ($SNLI_t$) is selected for tuning a model in Experiment 1. The reason for extracting the subset of the training data is the time and resource limitations for tuning the full *SNLI* dataset. The subset contains 41 000 items that have the most overlap with the vocabulary used in *AR* items. This size matches the *Veridicality* dataset described in subsection 3.5.4 so the results of the models trained on the two sets are comparable.

3.5.3 Training Data: Experiment 2

An *AR* training set of items from annual reports is used in Experiment 2. This data contains items that are very similar to the test data (subsection 3.5.1). A total of 1808 sentence pairs are annotated for training, following the same annotation scheme as for the *AR* test set. The sentence pairs are preprocessed following the same steps as for the test data. Nonetheless, there are a few differences between the datasets.

Firstly, the data covers earlier years of the annual reports, which could result in some style differences due to time. However, this division is selected since it makes a realistic setting where one would attempt to extract information from the most recent annual reports given the events in the earlier reports. Secondly, the training items are restricted to only sentences that contain both a mention of an aspect of SDGs and any form of a word from the set of ['reduce', 'less', 'drop', 'minimize', 'decrease', 'lower', 'fall', 'cut', 'shrink', 'decline', 'deflate']. With this restriction there are less items in the training set that do not concern reduction, as compared to the test set. This restriction was made in order to have more 'entailment' class items in the training set of *AR* without having to annotate a very large number of sentences.

3.5.4 Training Data: Experiment 3

In Experiment 3, various event factuality corpora (FactBank [Saurí and Pustejovsky, 2009], It Happened [White et al., 2016], UW [Lee et al., 2015]) as well as one treebank without factuality labels (*Penn Treebank (PTB)* [Marcus et al., 1994]) are converted to the textual entailment data format and used for training an RTE model. The event factuality corpora contain items in the format of example 21, which are recast to the sentence pair format in example 22.

(21) SENTENCE BLOCKBUSTER ENTERTAINMENT CORP. *said* it raised $92 million from an offering of liquid yield option notes.
EVENT WORD raised
LABEL underspecified
[Saurí and Pustejovsky, 2009]

(22) PREMISE BLOCKBUSTER ENTERTAINMENT CORP. *said* it raised $92 million from an offering of liquid yield option notes.

HYPOTHESIS it raised $92 million from an offering of liquid yield option notes.

In addition, even though the PTB treebank does not contain factuality labels, the sentences also contain mentions of events and thus items from this treebank have also been recast and used in Experiment 3. The factuality judgments for these items are added based on a pre-defined list of non-veridical items (see example 4.4 for details).

Sentences from the event factuality corpora are treated as premises, whereas hypotheses are generated by extracting the event-denoting spans from the premises. The labels are converted to the textual entailment format as well. The 'non-factual' label for example 21 is replaced by the 'no entailment' label for example 22. The full process of making the format and the annotations of the different corpora uniform is laid out in Chapter 4. The separate chapter is devoted to the description of the algorithm as it presents the generation of a new dataset.

All these recast corpora together compose the *Veridicality* dataset (*Ver*) that is used for training in Experiment 3. Table 1 presents the sizes of the resulting recast corpora and an estimate of correctly labeled items from each of them.[6]

Corpus	Size	Accuracy
FactBank	9301	94%
It Happened	24198	87%
UW	11026	94%
PTB	5715	93%

Table 1: Accuracy of annotation for training items

3.5.5 Training Data: Experiment 4

The data used in Experiment 4 (*Ver**) is the same as in Experiment 3 (*Ver*), except the items in *Ver** have random word insertions. The random words are added in order to make words in the premise and the hypothesis in a pair overlap less. I insert 8 random words in random positions of the premise and 2 random words in random positions in the hypothesis in all items in the *Ver* dataset. The number of random words is selected on the basis of the fact that the premises are in most cases much longer than the hypotheses. For example, the premise a. and hypothesis a. in example 23 from the *Ver* dataset become the premise b. and the hypothesis b. in example 23 in the new *Ver** dataset. The inserted random words are italicized.

[6]The accuracies are estimated by hand annotating 100 randomly selected items from each corpus. Only items in which the assigned label is wrong are considered false, excluding the items in which there are subsentence extraction mistakes if they do not lead to incorrect labels.

(23) PREMISE a. Hadson Corp. said it expects to report a third-quarter net loss of $17 million to $19 million because of special reserves and continued low natural-gas prices. [Marcus et al., 1994]

HYPOTHESIS a. it expects to report a third-quarter net loss of $17 million to $19 million because of special reserves

PREMISE b. Hadson Corp. *Yeargin* said it expects to report a third-quarter net loss of *abnormally scents* $17 *in-store* million to $19 million *helpful stimulating Attorney* because of *cradle-to-gate* special reserves and continued low natural-gas prices.

HYPOTHESIS b. it expects to report *infiltrating* a third-quarter net loss of *usurp* $17 million to $19 million because of special reserves

3.6 Baseline

The baseline for all four experiments described in the Chapters 5, 6, 7 and 8 is set by attributing all instances to the largest class, namely 'no entailment'. This sets the baseline at 89.96% accuracy as there is a high class imbalance in the test set. That is, most of the sentences in the annual reports, even if they mention emissions or pollution, do not actually report positive changes with regard to the SDGs. This accuracy, however, is not a realistic estimate for the baseline, because when attributing everything to the largest class, no true positives for the 'entailment' class are produced. A weighted F_1 score is better suited for the baseline as it takes into account the precision, recall and the imbalance of the classes. Hence, the baseline is an F_1 score of 84.78%.

4 A Method for Recasting Training Data

In this chapter I describe one of the contributions of this thesis, which is the algorithm for recasting event factuality data as RTE data. This process requires the extraction of the part of a given sentence which refers to the event of interest (the *subsentence*). The subsentence serves as the hypothesis and the original sentence as the premise in the RTE format. The subsentences are sought for by using dependency parses produced by CoreNLP [Manning et al., 2014] and the algorithm described in this chapter. The algorithm is built taking into account the annotation guidelines of each corpus as well as the Universal Dependency Relation descriptions [De Marneffe et al., 2014].

First, the *event denoting verb* and its dependents comprise the main part of the subsentence.[7] Second, the phrase describing the *agent-like entity* of the event is used as the subject of the subsentence. The agent is not always found among the dependents of the event denoting word as it is not always the subject of the event predicate. For example, the event of 'swimming' in example 24 is described not only by the event word 'swimming' but also by the agent 'she', even though 'she' is not the subject of the event verb. Dependents of the agent are also included into the subsentence.

(24) She changed for swimming.

The procedure for finding the agent description in these cases is elaborated on in the following sections.

4.1 Recasting *FactBank* to entailment format

To begin with, the event denoting word and its dependents are extracted. This process is the same for every corpus. In contrast, the task of finding the agent-like entity in the event is more complex and requires additions to the algorithm when handling the different corpora. Here I present the rules for *FactBank*, which are also used for recasting the remaining corpora.

In case the event word is a noun, no agent-like entity is needed as the noun phrase defines the event fully. For instance, the word 'offering' together with its dependents (the subsentence in example 25) denotes the offering event from the full sentence in example 25.

(25) SENTENCE BLOCKBUSTER ENTERTAINMENT CORP. said it raised $92 million from an *offering* of liquid yield option notes.
SUBSENTENCE an offering of liquid yield option notes

[7] The dependents that describe the modality or other attributes external to the event itself are not included. The types of the excluded dependents are: the adverbial modifiers, (subordinating) conjunction, parataxis, auxiliary and negation dependencies of the event denoting word.

Otherwise, if the event word is a verb or an adjective, the phrase denoting the agent-like entity for the event is searched for. Table 2 illustrates the rules for finding the agent-denoting word for the events.[8]

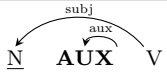
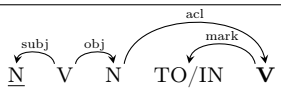
#	Type	Example	Parse	Subsentence
1.	Matrix verb	She **swims**.		She swims
2.	Copula or auxiliary verb	She **is** swimming		She is swimming
3.	Infinitival clausal modifier	She has the right to **swim**		She swim
4.	Infinitival adverbial modifier	She changed for **swimming**		She swimming

Table 2: Algorithm for finding the subsentence subject

Each rule describes how to find the agent phrase given the syntactic type of the event phrase. The word in bold in the example column is the *event denoting word* and the equivalent bolded node in the parse column is its position in a the syntactic structure. The underlined word is the agent entity and the underlined node is where it would be found in the dependency parse.

The rules in table 2 are applied in succession – the syntactic structure that embeds the event word in the sentence from the corpus is compared to the parse pattern in each rule until one of them matches. If a match is found, the item in the underlined position is treated as the agent for the subsentence.

In case 1. the agent is simply the subject dependent of the event word.[9] In case 2. the agent is the subject dependent of the parent of the copula or auxiliary event word. If the event word is a verb in an infinitival clausal modifier or an infinitival adverbial modifier, the subject in the underlined position in row 3. or row 4. is interpreted as the agent entity for the event.

The rules are applied recursively to deal with cases in which the event word is embedded under more than one of the structures in the table. For instance, the parse which embeds the event word could match the parse in one of the rules up to but not including the underlined subject node. The parse besides the subject dependent is then collapsed into a single node and treated as the new bold node. The parse embedding this new bold node is then again matched against all the parses in the table. For example, in a sentence such as "She

[8]The parses in the table are simplified and do not include nodes that are not relevant for the agent extraction procedure.

[9]More complex dependency structures than the given example can also be solved with the parse in row 1. For example, enhanced dependencies include subject dependencies for controlled verbs in open clausal complements.

changed to have the right to swim" the agent 'she' in the 'swimming' event can be found with the recursion rule.

The event denoting phrase preceded by the agent phrase compose the hypothesis for the sentence of the corpus. In addition to recasting the items into the sentence pair format, the labels have to be changed as well. The *FactBank* corpus contains multiple labels such as 'probable', 'possible', 'underspecified', etc. All of the labels are mapped to the 'no entailment' class, except for the 'certain' label, which is substituted with 'entailment'.[10]

4.2 Recasting *UW* to entailment format

For finding the spans of subsentences in the *UW* dataset [Lee et al., 2015], the items in the corpus are also parsed with enhanced dependencies. Thus, the same rules as the ones described in section 4.1 are used to find the agent phrase for the subsentences.

The labels for the events in this corpus are integers on the scale from -3 ('certainly did not happen') to 3 ('certainly happened') and every item is annotated by more than one person. Any event that is marked with a score higher than 2.5 on average gets the label 'entailment', and the remaining items get the label 'no entailment'. [11] For instance, the event denoted by 'become' in the premise in example 26 has a score of 2.6 in the *UW* corpus. In the new format, the sentence pair in example 26 is labeled with 'entailment'.

(26) PREMISE The program also calls for coordination of economic reforms and joint improvement of social programs in the two countries, where many people have become impoverished during the chaotic post- Soviet transition to capitalism. [Lee et al., 2015]

HYPOTHESIS many people become impoverished

4.3 Recasting *It Happened* to entailment format

The process of converting the *It Happened* corpus [White et al., 2016] into a sentence pair format is similar to that of *FactBank* described in section 4.1, with only a few differences. An advantage of the *It Happened* corpus is that gold parses are available for this dataset from Universal Dependencies [De Marneffe et al., 2014]. This improves the accuracy of the conversion process, as parsing errors are rare in manually parsed sentences. However, the corpus is parsed with basic dependency relations, which do not encode all the subtle relations that were present in the enhanced dependency relations used for *FactBank*.

Some adjustments to the procedure in section 4.1 had to be made due to the different parses. For instance, the basic dependency relations do not connect

[10] An exception is added to correct for the pitfall of the *FactBank* annotation scheme, in which events in the future tense can be classified as facts. If the *event denoting word* has a dependent 'will', the item is assigned the 'no entailment' label.

[11] I test various thresholds for splits of this scale into two labels, manually annotating the accuracy of the resulting labels attributed to the sentence pairs.

conjunct predicates to their subject. Hence, in a sentence such as example 27 there is no subject relation between 'you' and the predicate 'crossed'.

(27) Let me put it this way - If you have watched TV recently, crossed a modern bridge, flown in an airplane, received any form of modern medicine, did any mathematics, drove a car, used a cellular phone, etc. then you have somehow directly reaped the rewards which NASA has returned to us. [White et al., 2016]

Instead, the verb 'crossed' and the remaining coordinated verbs are only represented as the dependents of the preceding coordinated verb, such as 'watched'. Hence, rule 5. is added, which finds the subject of such coordinated verbs (see table 3). Just like all the rules, this rule is applied recursively, so that the subject dependent could be found for the all the coordinated verbs up to 'used' in sentence 27, by merging the verbs into a single node until rule 5. applies.

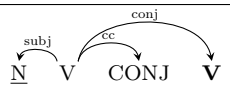
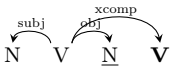
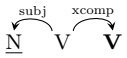
#	Type	Example	Parse	Subsentence
5.	Conjunction	She surfs and **swims**		She swims
6.	Open clausal complement (object control)	They let her **swim**		her swim
7.	Open clausal complement (subject control)	She decided to **swim**		She swim

Table 3: Additional rules for finding the subsentence subject

Similarly, the basic dependencies do not mark the subject relation between a subject or object controlled verb and the controlling entity. Therefore, rules 6. and 7. are added to be able to find the subsentence subject for the event verbs in the open clausal complements.

The annotation of *It happened* differs from the other corpora as well. In the case of *It happened* the annotators are asked to choose between the values 'True' and 'False' for an event predicate in a sentence, and provide their confidence level on a scale of 0 to 4. Every sentence and event pair is evaluated by two annotators. If the average confidence of the two annotators for a sample is 3 or higher and both annotators select the same value, the equivalent value ('entailment' for 'True' and 'no entailment' for 'False') is attributed to the sentence pair. In all other cases the value of the sentence pair is 'no entailment'. Hence, the confidence values are used to make a high threshold for the entailment class, so that the items include as many true entailments as possible.

4.4 Recasting *PTB* to entailment format

Finally, I investigate whether items for the *Veridicality* set could be created from a corpus that does not have factuality judgments. I use the *Penn TreeBank*

(PTB) [Marcus et al., 1994] for this exploration. Since the *PTB* is manually constituency-parsed, parsing errors are rare and the parses only need to be converted to dependency parse format, which is done using CoreNLP [Manning et al., 2014].

I treat each verb in the sentences in the *PTB* as an event, and check whether it is hedged by a potentially non-veridical operator. For each predicate in a given sentence, the same procedure as in Section 4.1 is followed in order to find its dependents and agent. The positions of subordinating verbs and the dependents of the event denoting verb are checked for the non-veridical items. If a potentially non-veridical item[12] hedges this event, the subsentence denoting this event and the original sentence are retrieved as a sentence pair for the *Veridicality* set. For instance, the event denoted by the copula ''s' is hedged by negation in the premise in example 28. The hypothesis denoting the event and the original sentence are thus added to the *Veridicality* set.

(28) PREMISE It's *not* a zero-sum game. [Marcus et al., 1994]
HYPOTHESIS It's a zero-sum game.

All sentence pairs extracted this way were assigned a 'no-entailment' label due to the fact that they have been generated by finding subsentences that are embedded in non-veridical contexts.[13]

[12] A pre-defined list of 83 non-veridical items is used, including negation, propositional attitude verbs, adverbs, connectives, modals, etc.

[13] Note that this is not a fool-proof rule, since a combination of non-veridical contexts can cancel each other out. However, the method of recasting sentences without event factuality labels appears to be scalable based on the high accuracy (93%) of the sentence pairs constructed and labeled this way.

5 Experiment 1

The main goal of this experiment is to establish how well an untuned and a tuned version of a neural RTE model trained on general NLI data performs on a specialized domain set and what that model learns about the linguistic phenomena in the data.

In all experiments (Chapters 5, 6, 7 and 8), italicized names of corpora are used to refer to the *datasets* themselves, while the names in bold are used to refer to the **models** trained on the respective datasets.

5.1 Setup

Two versions of the *SNLI* data are used to train two models. The first model is trained on the full *SNLI* training set similarly to the [Rocktäschel et al., 2015] model, only replacing the three class distinction with two classes ('entailment' and 'no entailment'). This model will be referred to by the **SNLI** name and only serves as the lower baseline of how a general RTE model performs on a specialized dataset.

The second model, trained on a subset of *SNLI* ($SNLI_t$), will be referred to as the $\mathbf{SNLI_t}$ model. By virtue of the training data being more manageable in size, this model is tuned to the *AR* development set. The tuning is important to the performance of the model, as the right combination of parameter values can improve it significantly. This setting serves to show how well a model can perform on a specialized test set having seen only a part of SNLI items in training, but having been tuned with a development set that closely resembles the test set. This tuned $\mathbf{SNLI_t}$ model will be used instead of **SNLI** for comparisons and combinations with other models in the remaining 3 experiments.

The test data for all 4 experiments is the specialized-domain Annual Report *AR* test set.

5.2 Results

Table 4 presents the baseline defined in section 3.6 and the results of the models in this experiment.

Model	Training size	Accuracy (dev)	Accuracy (test)	Weighted F_1 (test)	lr	l_2	p	Epochs
Baseline	N/A	N/A	89.96	84.78	N/A	N/A	N/A	N/A
SNLI	550k	N/A	47.76	57.09	1×10^{-3}	0	0.2	40
$\mathbf{SNLI_t}$	41k	87.16	86.11	83.65	1×10^{-3}	0	0	32

Table 4: Classification Results: Experiment 1

As expected, the **SNLI** model performs poorly with an F_1 score of 57.09, as it has seen no items resembling the annual report style in its training or development sets. In contrast, the $\mathbf{SNLI_t}$ model is tuned towards the *AR* development set and reaches an F_1 score of 83.65. Therefore, even though the

training set for this model is smaller by a factor of 13, it outperforms **SNLI** significantly[14]. $\mathbf{SNLI_t}$ comes very close to the largest class baseline, however it does not outperform it.

5.3 Analysis

When it comes to analyzing what information the model uses for making the classification decisions, the attention mechanism provides some insight about the semantic relations that the model selects as important. The attention visualizations show how much attention was paid to each word in the premise when processing each word in the hypothesis.[15] The lighter shades in the grid indicate higher attention weights.

It is evident that the **SNLI** model is not able to figure out which words to pay attention to in order to classify the items. For example, figure 1 illustrates the word-by-word attention scores for the premise "In addition, fifty bird boxes and nine water basins were installed." and the hypothesis "Water was reduced".

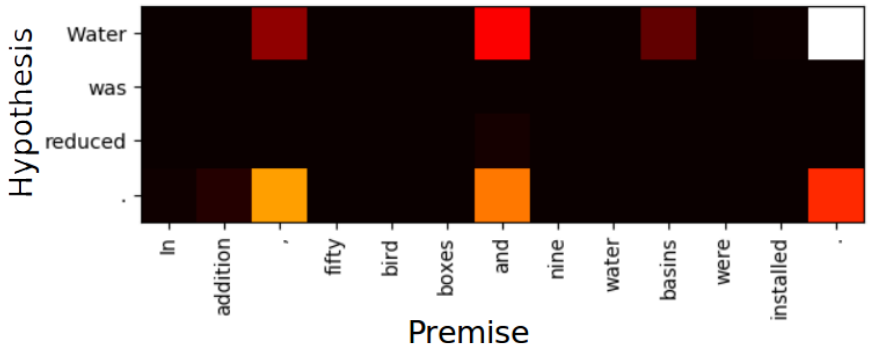


Figure 1: Attention weights in test item [Wereldhave Annual Report, 2017], **SNLI** model

The only relation between two content words that the model pays a little attention to is the thematically related 'water' and 'basin'. As discussed in the analysis of the original model [Rocktäschel et al., 2015], the model paying most attention to function words and punctuation is indicative of the attention mechanism not functioning properly. That is, the final classification decision probably does not depend on the words that were attended to, but rather the overall representation of the sentence pair.

[14]Significance is assessed using the t-test with SciPy [Jones et al., 2001]. The significance levels are as follows: * = $p < 0.1$, ** = $p < 0.05$, *** = $p < 0.01$.

[15]The visualization of the attention weight distribution is implemented in Matplotlib [Hunter, 2007].

In contrast, the $\mathbf{SNLI_t}$ model, which is tuned towards the *AR* development data, uses more information about relations between the words in the premise and the hypothesis. Figure 2 illustrates the word-by-word attention scores for the sentence pair of "The assessment covers a range of environmental issues, including evaluating energy and water use, health and wellbeing, pollution, transport, materials, waste, ecology and managemenet processes" and "Waste was reduced".

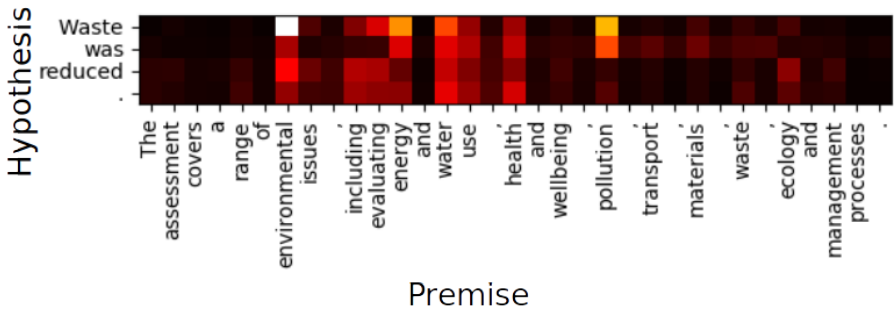


Figure 2: Attention weights in test item [PostNL Annual Report, 2017], $\mathbf{SNLI_t}$ model

The attention visualization shows that the model pays attention to multiple words in the premise from the environmental topic ('environmental', 'energy', 'water', 'pollution') especially when processing the word 'waste' in the hypothesis. This shows that the model relies on the similarity of topic between the two sentences. However, the model does not pay attention to the word 'waste' in the premise when processing the same word in the hypothesis. This hints that the model does not treat identity as similarity when learning about entailment, which I examine further.

An additional artificial set of 31 identical sentence pairs is constructed to test whether the model indeed does not recognize identity as a relevant relation for evaluating similarity. The set consists of premises extracted from *SNLI* and hypotheses that are identical to the premises. The results support the said interpretation as $\mathbf{SNLI_t}$ reaches only 12.9% in accuracy on this small set.

Furthermore, the $\mathbf{SNLI_t}$ model learns to recognize the relations between words that often appear together. To illustrate, figure 3 shows the word-by-word attention scores for the sentence pair "We utilize inherently high energy processes and focus strongly on reducing carbon footprint and energy use, while saving costs in our own operations" and "Footprint was reduced".

The word 'reduced' often takes noun phrases such as 'energy' and 'energy use' as its arguments. The $\mathbf{SNLI_t}$ model recognizes this relation as it attends to the phrase 'energy' and 'energy use' in the premise when processing the word 'reduced' in the hypothesis. Hence, it can be argued that the model learns

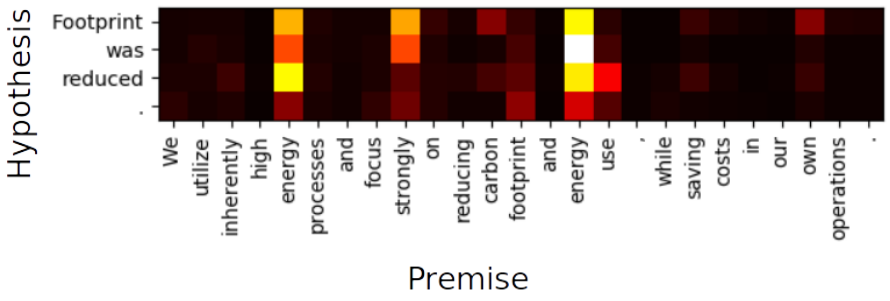


Figure 3: Attention weights in test item [Akzo Nobel Annual Report, 2017], $\mathbf{SNLI_t}$ model

relations between words based on their word embedding representations, which encode information about the contexts of words.

In order to inspect what the model has learned about veridicality, an additional small artificial test set of *Non-Veridical Operators* (*NVO*) is constructed. The hypotheses for these items are taken from the *SNLI* test set. The premises are produced by inserting a non-veridical operator into that sentence. For instance, the hypothesis in example 29 appears in the test set of *SNLI*. Then, the premise in example 29 is produced by adding the non-veridical propositional attitude verb 'considers' to it. A total of 31 such sentences are produced with one non-veridical operator in each sentence.

(29) PREMISE A couple *considers* walking hand in hand down a street.
HYPOTHESIS A couple walk hand in hand down a street.

Both the $\mathbf{SNLI_t}$ and **SNLI** models are tested on the *NVO* set to see whether the models attend to the non-veridical operators when the remaining part of the sentence is identical. While **SNLI** performs very poorly on this dataset with an accuracy[16] of 16.13%, $\mathbf{SNLI_t}$ performs substantially better with an accuracy of 77.42 (see Table 5). This improvement could once again be attributed to the fact that the $\mathbf{SNLI_t}$ model is tuned to the development set that contains relevant examples, namely non-veridical contexts.

The attention visualizations for the **SNLI** model on the *NVO* set support the conclusion that the model does not learn much about non-veridicality. For example, when classifying the sentence pair in example 29, **SNLI** pays attention to all the content words in the premise when an identical word or another inflection of the same word appears in the hypothesis (see Figure 4). **SNLI** does

[16] Accuracy is used in this test, as the F_1 score is not meaningful without any true positive items (items in the 'entailment' class). No 'entailment' examples are present because the small dataset is composed exclusively of items that capture the effect of non-veridical contexts.

Model	Training size	Accuracy (NVO)
SNLI	550k	16.13
SNLI$_t$	41k	77.42

Table 5: Classification Results: *NVO* test set

not pay any attention to the non-veridical operator 'considers' in the premise. This suggests that the model does not have any representation of the non-veridicality of the hedge 'considers'. However, the fact that **SNLI** attends to identical words points out that perhaps the larger training set (as compared to $SNLI_t$) allows the model to learn that identity and not only similarity is relevant for entailment.

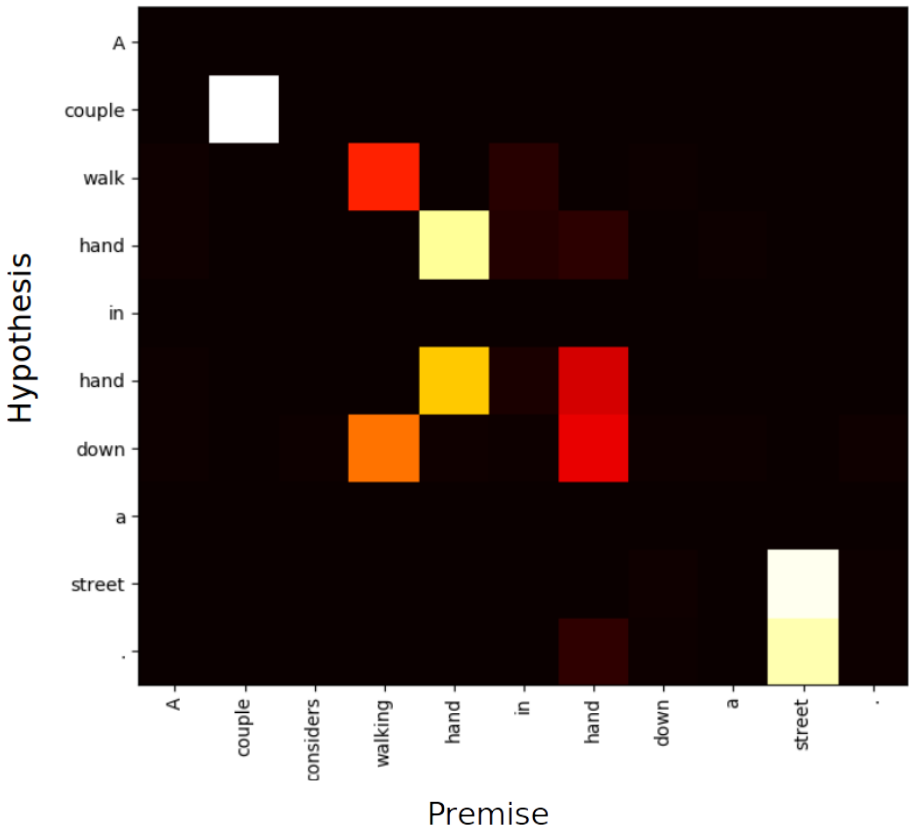


Figure 4: Attention weights in test item, **SNLI** model

On the other hand, the **SNLI$_t$** model occasionally pays attention to the more common non-veridical operators and the predicates that they hedge. For

example, the model figures out the relationship between the non-veridical 'try' in the premise "3 young men in hoods try standing in the middle of a quiet street facing the camera" and its subordinate verb 'standing' when it appears in the hypothesis (Figure 5).

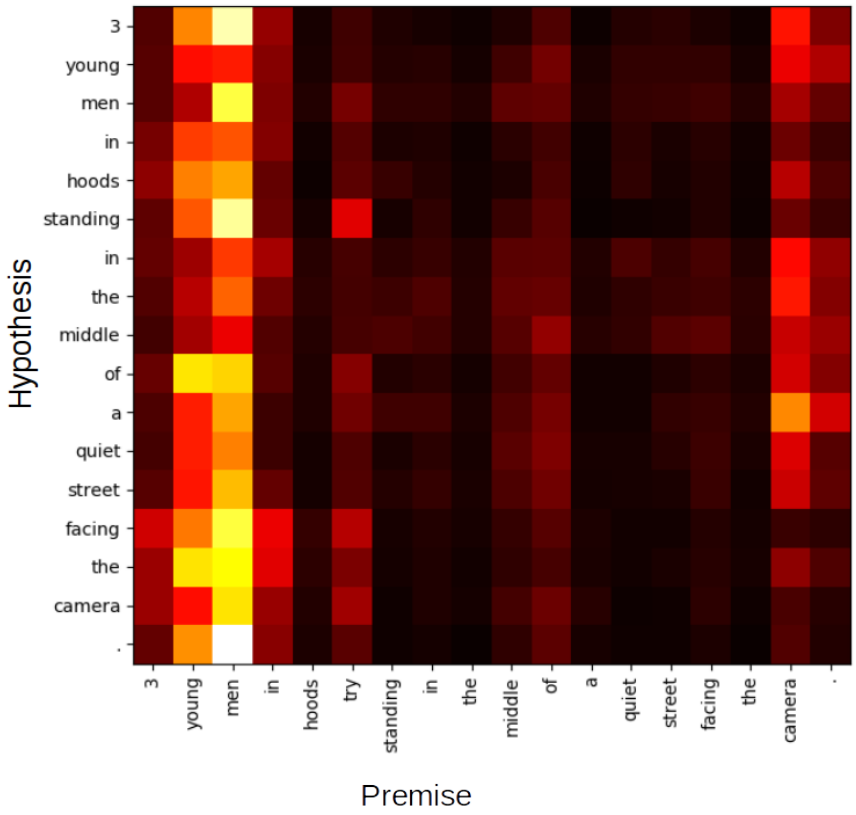


Figure 5: Attention weights in test item, $\mathbf{SNLI_t}$ model

All in all, it appears that a model trained on the $SNLI_t$ dataset (subset of *SNLI*) is able to learn the notion of veridicality to some extent, provided that it is tuned to the development set in which non-veridical operators are relevant for classification. In addition, the results of the **SNLI** and $\mathbf{SNLI_t}$ show that the models are not able to perform better than the largest class attribution baseline when the domain of the test set is rather specific. In order to see whether the model can be improved with regard to this particular domain by adding some domain-specific data to the training, Experiment 2 is conducted.

6 Experiment 2

This experiment is carried out in order to find out whether adding domain-specific data to the training set of a neural RTE model could improve its performance on an annual report test set. In addition, Experiment 2 evaluates what this model learns from the annual report data in contrast to the general NLI data.

6.1 Setup

The *AR* training dataset was used on its own in training the **AR** model as well as in combination with the $SNLI_t$ dataset to train the **AR+SNLI$_t$** model. Both these models were tuned to the *AR* development set and tested on the *AR* test set.

6.2 Results

The results of the **AR** and **AR+SNLI$_t$** models are presented in the highlighted section in Table 6 along with the performance of the previous models.

Model	Training size	Accuracy (dev)	Accuracy (test)	Weighted F_1 (test)	lr	l_2	p	Epochs
Baseline	N/A	N/A	89.96	84.78	N/A	N/A	N/A	N/A
SNLI	550k	N/A	47.76	57.09	1×10^{-3}	0	0.2	40
SNLI$_t$	41k	87.16	86.11	83.65	1×10^{-3}	0	0	32
AR	2k	83.14	82.20	84.44	1×10^{-3}	0	0	29
AR+SNLI$_t$	43k	90.08	89.90	86.01***	1×10^{-4}	1×10^{-3}	0.2	21

Table 6: Classification results: Experiment 2

The **AR** model with an F_1 score of 84.44 comes very close to the baseline but does not outperform it. This could perhaps be explained by the very small size of the training set (1808 items) or the fact that the test data has much more variation than the training data. To be precise, the lack of variation in the *AR* training set can be attributed to the fact that it has an additional constraint for what kind of sentences are selected, namely only the ones that mention not only an SDG aspect but also some form of 'reduction'.

However, when the domain-specific data is merged with the generic textual entailment data, the **AR+SNLI$_t$** model performs significantly better than the largest class baseline, reaching an F_1 score of 86.01.

6.3 Analysis

The analysis of the attention patterns of the **AR** model shows that the model uses domain-specific information that is useful for this particular classification problem. For example, figure 6 demonstrates that the **AR** model pays attention almost exclusively to the words in italics in the premise: "Our *target* is to reduce

our product cradle-to-grave carbon footprint by *25-30*% per ton of sales between 2012 and *2020*, including the impact from VOC emissions".

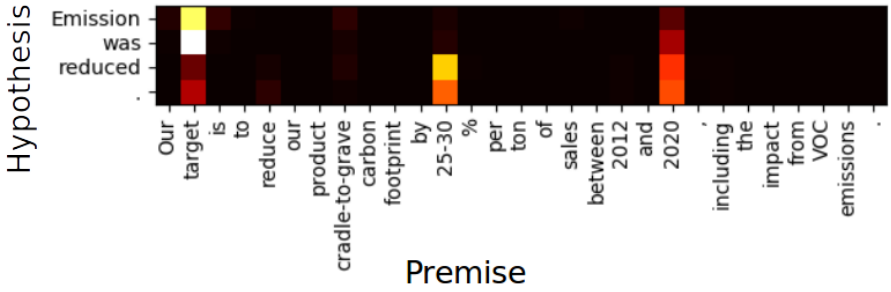


Figure 6: Attention weights in test item [Akzo Nobel Annual Report, 2017], **AR** model

The attention to the numbers 2020 and 25-30 suggests that the model infers the likelihood that sentences which refer the future do not usually report actual improvements, and that the actual accomplishments are often quantified. Many of these inferences that the model learns from the domain-specific data are not semantic or logical relations but only correlations that do not determine entailment relations definitively. For instance, in this case the quantification of 25-30% is mentioned without the reductions being actualized in the present.

Notwithstanding, the non-veridical items appear to be so common in the *AR* dataset, that the **AR** model learns to pick up on them. Namely, the model attends to the non-veridical 'target' in figure 6 and to the non-veridical 'committed' in the phrase 'committed to making progress' in figure 7. This indicates that the register used in the annual reports might be so uniform that the model is able to learn when phrases are hedged even from the small amount of data.

However, there is additional evidence that the **AR** model does not learn much about linguistic features. Firstly, the model does not attend to words with the same root in the premise and the hypothesis in figure 6, such as 'emission' and 'emissions', 'reduced' and 'reduce'. The way that the sentence pairs have been generated can account for this shortcoming of the model. The hypotheses in *AR* have been selected on the basis of containing the word of interest in the premise ('CO_2', 'water', etc.). Hence, it is not surprising that the model does not pay attention to words that always appear in the hypothesis whenever they appear in the premise. Therefore, the model could not be expected to correctly classify test items in which the target word did not repeat itself in both the premise and the hypothesis.

Secondly, the relevance of numerals and other hints is interpreted quite crudely by the **AR** model. That is, the words '2020' and 'target' in figure 6

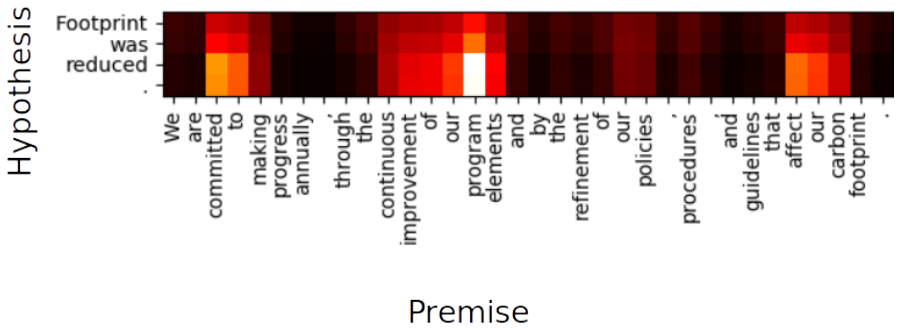


Figure 7: Attention weights in test item [ARCADIS Annual Report, 2017], **AR** model

are attended to throughout all the words in the hypothesis. This means that the model knows that 'target' is an important word but does not necessarily make the connection to the relevant event of reducing emissions that 'target' scopes over. This could be attributed to the lack of variation in the phrasing of the hypotheses in the training set of this model. That is, the hypotheses follow the same formulation in all examples, the only changing aspect being the target word. Hence, it is not surprising that the model does not learn to differentiate between the words in the hypothesis.

As far as the **AR**+**SNLI$_t$** model is concerned, strangely, it attends to all words equally. This could indicate that the model relies on the final representation of the sentence pair in LSTM to make the classification decision without taking the attention scores intro consideration.

Both the **AR** and **AR**+**SNLI$_t$** models are also tested on the small *NVO* set with a view to finding out what they learn about non-veridical operators. The results are presented in table 7.

Model	Training size	Accuracy (NVO)
SNLI'	550k	16.13
SNLI$_t$	41k	77.42
AR	2k	77.42
AR+**SNLI$_t$**	43k	90.32

Table 7: Classification Results: *NVO* test set

While the **AR** model performs as well as **SNLI$_t$** with an accuracy of 77.42%, when the two training sets are combined, **AR**+**SNLI$_t$** reaches 90.32% accuracy. This improvement signals that the former two models may be learning different non-veridical operators and thus the combination of the two training sets covers

a larger spectrum of hedges. Non-veridical operators such as 'target' are very common in the report genre, while others, such as 'want' are more usual in general discourse.

Overall, it appears that both the general and the specific datasets are necessary for the model to beat the baseline in classifying the items in the annual report domain. More general relations are learned from $SNLI_t$ as demonstrated in Experiment 1, and the domain-specific cues are learned from the AR dataset. While the models seem to learn lexical relations from the $SNLI_t$ corpus and idiosyncratic features from the AR dataset, the **AR+SNLI$_\mathbf{t}$** model trained on both datasets appears to acquire some knowledge of veridicality, as it performs well on the NVO set. However, since the **AR+SNLI$_\mathbf{t}$** model attends to all words equally, no analysis on its performance can be carried out. In order to see whether the model trained on the linguistically specialized *Veridicality* data can reach even higher and potentially more scalable results, Experiment 3 is conducted.

7 Experiment 3

The goal of this experiment is to test whether a neural RTE model can be improved by training it on linguistically specialized data that covers the phenomenon of veridicality. It also analyzes what linguistic cues the model pays attention to when trained on the *Veridicality* data, in order to establish what linguistic phenomenon the model is able to learn.

7.1 Setup

The recast *Veridicality* set is referred to as *Ver* and is used to train the **Ver** model. In addition to it, this training set is combined with the domain-specific *AR* dataset to train **Ver+AR**, as well as with the general $SNLI_t$ dataset to train **Ver+SNLI$_t$**, and finally with both datasets to train **Ver+AR+SNLI$_t$**. All three models are tuned to the *AR* development set and tested on the *AR* test set.

7.2 Results

Table 8 presents the results of Experiment 3 in highlight along the results of the previous experiments.

Model	Training Size	Accuracy (dev)	Accuracy (test)	Weighted F_1 (test)	lr	l_2	p	Epochs
Baseline	N/A	N/A	89.96	84.78	N/A	N/A	N/A	N/A
SNLI	550k	N/A	47.76	57.09	1×10^{-3}	0	0.2	40
SNLI$_t$	41k	87.16	86.11	83.65	1×10^{-3}	0	0	32
AR	2k	83.14	82.20	84.44	1×10^{-3}	0	0	29
AR+SNLI$_t$	43k	90.08	89.90	86.01***	1×10^{-4}	1×10^{-3}	0.2	21
Ver	51k	32.80	33.81	40.38	1×10^{-3}	0	0	32
Ver+SNLI$_t$	92k	37.16	35.88	42.89	1×10^{-3}	1×10^{-4}	0	33
Ver+AR	52k	87.74	86.57	87.05***	3×10^{-4}	0	0.1	24
Ver+AR+SNLI$_t$	94k	85.21	83.70	85.37***	1×10^{-3}	0	0.2	28

Table 8: Classification results: Experiment 3

The **Ver** model yields very low performance on the test data. This could be expected since the *Ver* data contains information about only one type of semantic phenomenon that could be responsible for the presence or absence of entailment between sentences. It does not, for instance, encode any information about synonymy or hyponymy. Even when the *Ver* data is combined with the $SNLI_t$ training data, the model **Ver+SNLI$_t$** reaches surprisingly low results.

However, when the *Ver* data is combined with *AR* data, the model **Ver+AR** performs better than any other model, even the one that is trained on all three datasets – **Ver+AR+SNLI$_t$**. Both **Ver+AR** and **Ver+AR+SNLI$_t$** outperform the largest class baseline significantly with an F_1 score of 87.05 and 85.37, respectively.

7.3 Analysis

When it comes to analyzing what the **Ver** model learns, attention visualization provides insights again. Figure 8 below and figure 9 on page 45 illustrate the word-by-word attention scores of this model for two test sentence pairs.

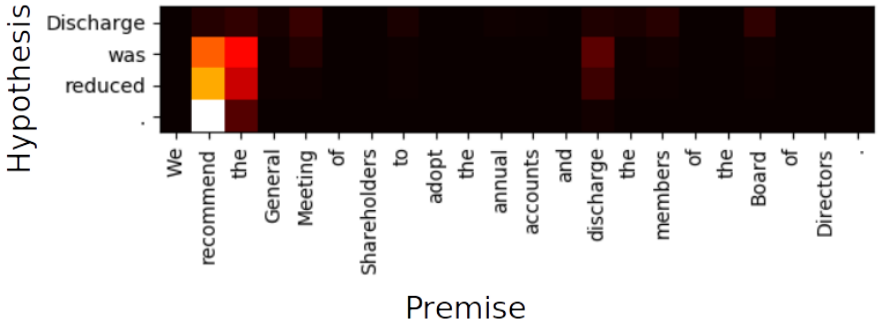


Figure 8: Attention weights in test item [X5 Retail Group Annual Report, 2017], **Ver** model

The words in italics in the following premises are mostly attended to: "We *recommend the* General Meeting of Shareholders to adopt the annual accounts and discharge the members of the Board of Directors" and "The Group recognizes a provision *if* the Group has an obligation to restore a leased asset in its original condition at the end of its lease term and in case of legal requirements with respect to clean-up of contamination of land, and the *estimate can* be made reliable". It can be seen that the model trained on *Ver* data learns to attend to the non-veridical contexts such as the object-control verb 'recommend', the conditional conjunct 'if' and the modal 'can'.

Similarly to the **AR** model, **Ver** does not differentiate between the words of the hypothesis very much. This suggests that the model is using the presence of the non-veridical operator for the classification decision regardless of what clause within the sentence it scopes over. The same pattern of lack of differentiation between the words in the hypothesis is also very prevalent in testing the **Ver**+**AR** model (see Figure 10, p. 47).

However, **Ver**+**AR** appears to build more sophisticated representations of non-veridicality than **Ver**. Figure 10 (p. 47) shows that the **Ver**+**AR** model pays attention to the non-veridical words as well as what they scope over in the premise "Figures and calculation methodology: Detailed environmental figures including intensity figures, targets and *avoided energy consumption* by our customers *can* be *found* in Appendix 6: Environmental figures" (attended words italicized). The model is able to learn not only to recognize the non-veridical items but also the words that denote the events which are made non-factual by

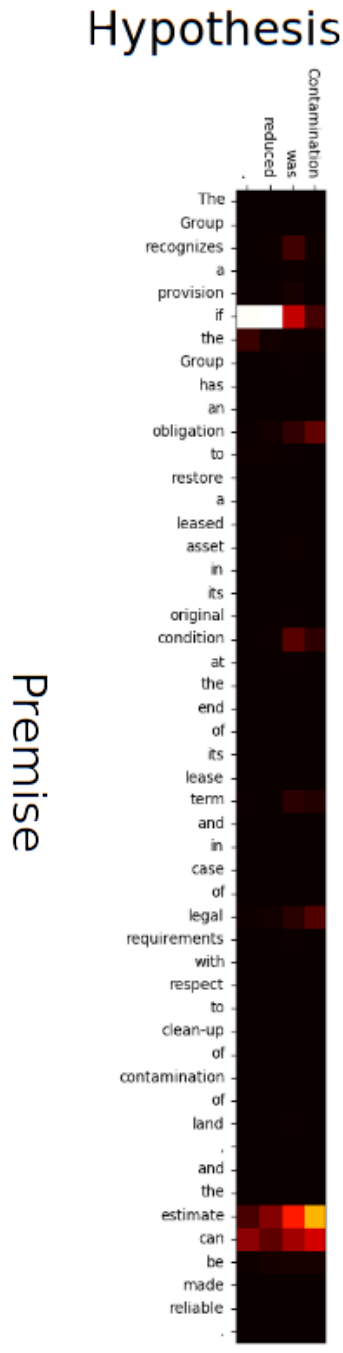


Figure 9: Attention weights in test item [OCI Annual Report, 2017], **Ver** model

these operators, such as the '(avoided) energy consumption'.

When all three sets of training data are combined, the model **Ver+AR+SNLI$_t$** attends to an even more varied range of phenomenon, albeit missing some important nuance. For example, the model attends to word pairs that refer to the same topic, such as 'water' in the hypothesis "Water was reduceed" and 'spill' in the premise "All product that was spilled into water was removed and the spill did not lead to a formal permit violation" (Figure 11, p. 48)). However, the model only makes the connection between 'spill' and 'water' but not 'spilled' and 'water'. Moreover, in the same example, the model attends to the (non-veridical) negated verb in the phrase 'not lead' in the premise, even though this non-veridical operator does not actually concern the content of the hypothesis.

Even though the models attend to the intended linguistic features, some of them perform not as well as it was expected. Thus, in order to see whether it has learned that some words are are non-veridical, the model is also tested on the *NVO* dataset. The results of all the models that included *Ver* in their training set are on the *NVO* test set are highlighted in Table 9.

Model	Training size	Accuracy (NVO)
SNLI'	550k	16.13
SNLI$_t$	41k	77.42
AR	2k	77.42
AR+SNLI$_t$	43k	90.32
Ver	51k	38.71
Ver+AR	52k	32.26
Ver+SNLI$_t$	92k	45.16
Ver+AR+SNLI$_t$	94k	38.71

Table 9: Classification Results: *NVO* test set

Surprisingly, it appears that all of the models in this experiment (**Ver**, **Ver+AR**, **Ver+SNLI$_t$**, **Ver+AR+SNLI$_t$**) perform very poorly on this specialized artificial dataset. One possible explanation for this unexpected result is the nature of the *Ver* and *NVO* datasets. The *Ver* dataset is semi-artificially produced and therefore it has a very limited range of sentence pairs. More precisely, the dataset contains only sentences in which the hypothesis is a literal rewrite of a part of the premise. In most cases the premise contains a non-veridical operator while the hypothesis does not, which explains why the model learns to attend to the non-veridical operator, as previously shown. However, the model never sees examples in training in which some words in the hypothesis do not appear in the premise (which is present in the *AR* test set), nor examples in which the premise has almost no additional words besides the non-veridical operator when compared to the hypothesis (which is the case in the *NVO* test set).

If this in fact is the reason why **Ver** does not perform well, it could also explain why **Ver+AR** reaches such high performance in the main test set. That is, since the *AR* dataset also contains many non-veridical operators, the *Ver+AR*

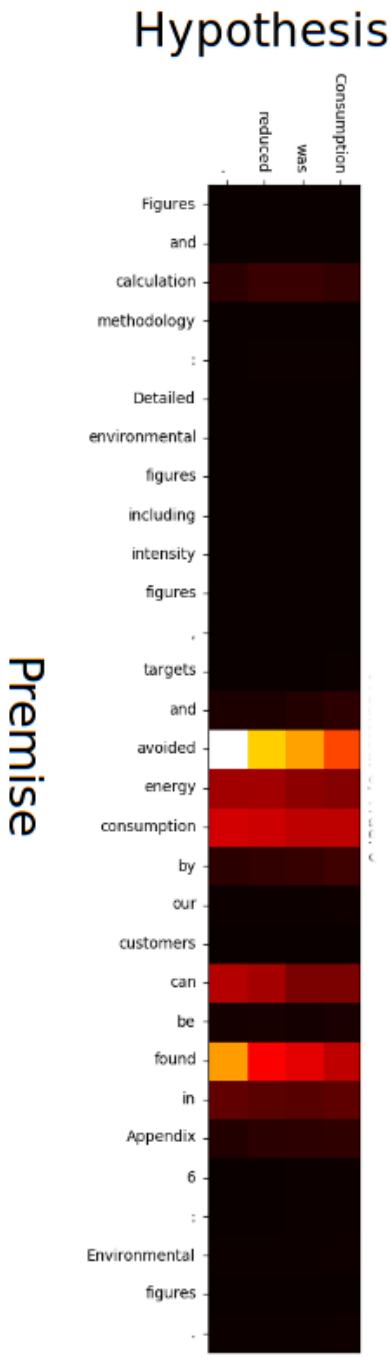


Figure 10: Attention weights in test item [KPN Annual Report, 2017], **Ver+AR** model

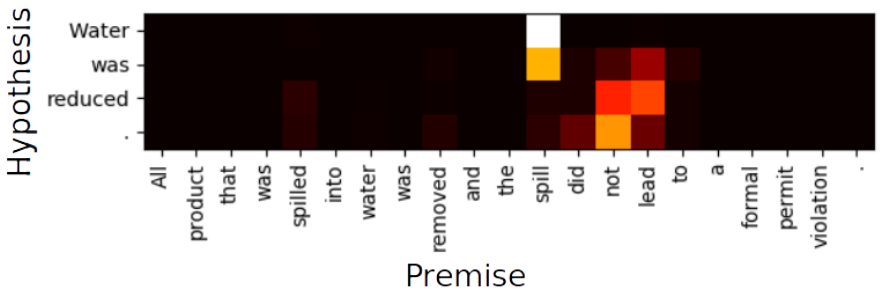


Figure 11: Attention weights in test item [Vopak Annual Report, 2017], **Ver**+**AR**+**SNLI$_t$** model

training set contains more varied examples of how non-veridical operators are used. The **Ver**+**AR** model is thus able to learn how non-veridical operators function.

However, the fact that the **Ver**+**AR**+**SNLI$_t$** model performs worse than **Ver**+**AR** and **AR**+**SNLI$_t$** is rather unexpected, since it has additional training data that has been effective in training a model in Experiments 1 and 2. One possible explanation for this lower result of **Ver**+**AR**+**SNLI$_t$** could be that the *SNLI$_t$* and *Ver* datasets are incompatible for training a model. Such an interpretation is also supported by the very low performance of the **Ver**+**SNLI$_t$** model. The incompatibility of the **SNLI$_t$** and **Ver** models could be attributed to *SNLI$_t$* and *Ver* datasets having very different sentences, which cover distinct linguistic phenomena. While the *Ver* dataset contains many non-veridical operators, it does not contain any synonyms or paraphrases between the premise and the hypothesis. On the other hand, *SNLI$_t$* contains synonyms and taxonomic relations, yet it does not contain any sentence pairs that illustrate how non-veridical operators are used. Therefore it is possible that **Ver**+**SNLI$_t$** performs poorly on examples that contain both non-veridical operators and paraphrases in the same sentence pair, as the model never sees such examples at training time. The *AR* test set is comprised of exactly such sentences, containing many paraphrases as well as non-veridical operators.

The attention scores for the sentence pairs tested with **Ver**+**SNLI$_t$** support the interpretation that the two datasets are incompatible. The model does not attend to the lingusitically relevant items that **Ver** or **SNLI$_t$** attend to. Instead, it attends only to some words in the premise and shows the least differentiation between the words in the hypothesis of all the models in the 3 experiments. For example, figure 12 on page 50 presents the attention scores for the premise "The company separates waste at the source and works with secondary parties who specialize in sorting and recycling waste". The attended phrases 'waste' and

'recycling waste' relate to the hypothesis, however the two phrases that get the most attention ('source' and 'works with secondary parties who') are common expressions in industry reports and are not specifically related to the question of waste reduction. These attention scores vary only minimally with regard to the words in the hypothesis. This could mean that the model does not take the hypothesis into account and instead uses cues in the premises to determine whether the item is of the $SNLI_t$ type or the *Ver* type, as the training items are so different.

By and large, the RTE model is able to learn about non-veridicality when trained on both the domain and the linguistically specialized data. While the *AR* data provides the examples of how non-veridicality is used in the particular annual report register, the *Ver* data supplies more numerous examples of how non-veridical items are used in general. However, the results of the experiment indicate that the linguistically specialized recast *Veridicality* data is not compatible with the generic *SNLI* data.

In order to test whether the models in Experiment 3 are not performing as well as expected due to the homogeneity of their training data, experiment 4 is carried out.

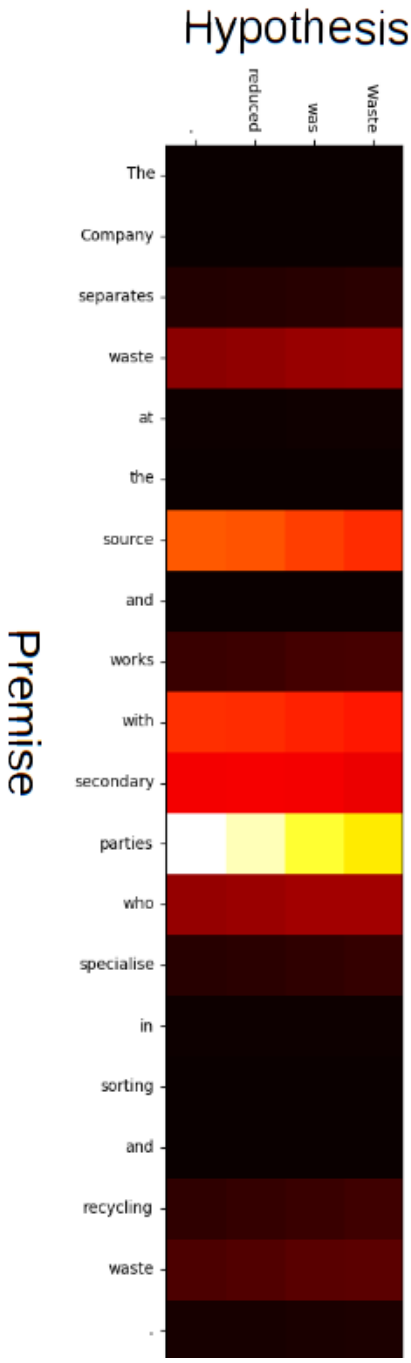


Figure 12: Attention weights in test item [Euronext Annual Report, 2017], **Ver+SNLI$_t$** model

8 Experiment 4

This experiment is designed to test whether the rigid composition of the sentences in the training set of Experiment 3 are the cause of the low performance of the models in that experiment. Hence, this experiment tests how inserting random words affects the performance of the models trained on the *Veridicality* data.

8.1 Setup

The models **Ver***, **Ver***+**AR**, **Ver***+**SNLI$_t$** and **Ver***+**AR**+**SNLI$_t$** are trained on the respective datasets for a comparison with the equivalent models without the random insertions, namely **Ver**, **Ver**+**AR**, **Ver**+**SNLI$_t$** and **Ver**+**AR**+**SNLI$_t$**. All of the models are tuned to the *AR* development set and tested on the *AR* test set.

8.2 Results

The results of all 4 experiments are presented in table 10. The results of Experiment 4 are highlighted.

Model	Training Size	Accuracy (dev)	Accuracy (test)	Weighted F_1 (test)	lr	l_2	p	Epochs
Baseline	N/A	N/A	89.96	84.78	N/A	N/A	N/A	N/A
SNLI$_t$	41k	87.16	86.11	83.65	1×10^{-3}	0	0	32
AR	2k	83.14	82.20	84.44	1×10^{-3}	0	0	29
AR+**SNLI$_t$**	43k	90.08	89.90	86.01***	1×10^{-4}	1×10^{-3}	0.2	21
Ver	51k	32.80	33.81	40.38	1×10^{-3}	0	0	32
Ver+**SNLI$_t$**	92k	37.16	35.88	42.89	1×10^{-3}	1×10^{-4}	0	33
Ver+**AR**	52k	87.74	86.57	87.05***	3×10^{-4}	0	0.1	24
Ver+**AR**+**SNLI$_t$**	94k	85.21	83.70	85.37***	1×10^{-3}	0	0.2	28
Ver*	51k	60.69	59.59	67.45	1×10^{-3}	1×10^{-4}	0	25
Ver*+**SNLI$_t$**	92k	59.85	60.05	67.81	1×10^{-3}	1×10^{-4}	0	31
Ver*+**AR**	52k	81.07	80.65	83.08	1×10^{-3}	0	0.2	36
Ver*+**AR**+**SNLI$_t$**	94k	81.07	81.80	83.99	1×10^{-3}	0	0	28

Table 10: Classification results: Experiment 4

The model **Ver*** performs significantly better than the **Ver** model with the F_1 score of 67.45 and the **Ver***+**SNLI$_t$** model performs significantly better than the **Ver**+**SNLI$_t$** model, reaching an F_1 score of 67.81. These results confirm the utility of the randomization. The performance is still not very high, however this is not unexpected since the *Veridicality* data only encodes one type of phenomenon, as previously mentioned.

On the other hand, when the *Ver** or the *Ver*+SNLI$_t$* datasets are combined with the *AR* dataset in training, the improvements of random word insertion are not present anymore. That is, the **Ver***+**AR**+**SNLI$_t$** model performs worse than the **Ver**+**AR**+**SNLI$_t$** and **Ver***+**AR** performs worse than **Ver**+**AR**.

8.3 Analysis

The higher performance of the **Ver*** and **Ver*+SNLI$_t$** models as compared to the versions of the models without inserted random words is expected. It suggests that the randomization makes the model depend less on the overlap between the premise and the hypothesis in the training set, which is hypothesized to have caused the low performance of the **Ver** and **Ver+SNLI$_t$** models. However, the fact that the performance of the other models in this experiment do not improve suggests that something else needs to be accounted for.

The fact that **Ver*+AR+SNLI$_t$** performs worse than the **Ver+AR+SNLI$_t$** and **Ver*+AR** performs worse than **Ver+AR** could be accounted for by the contribution of the *AR* dataset. The *AR* data includes both non-veridical items and synonymous phrases. The presence of this data in the training set could be the reason why adding artificially paraphrased veridicality sentences into the training of **Ver*+AR+SNLI$_t$** and **Ver*+AR** does not benefit the models any more – they have already seen linguistically varied input. The fact that the **Ver*+AR+SNLI$_t$** and **Ver*+AR** models perform worse than **Ver+AR+SNLI$_t$** and **Ver+AR** could then be explained by the fact that adding the random words into the training set introduces noise and does not offer much information that the *AR* data was not already supplying. This interpretation is consistent with the results of Experiment 3, as it supports the conclusion that while *Ver* and *SNLI$_t$* datasets are not compatible, the *AR* dataset is compatible with both of the latter sets. Notwithstanding, the *AR* set is small and therefore cannot correct for the incompatibility of the *Ver* and *SNLI$_t$* sets in training **Ver+AR+SNLI$_t$** in experiment 3.

By and large, the experiment confirms the hypothesis that the lack of variation in the recast data is inhibiting the performance of the models in Experiment 3. In addition, it shows that inserting random words into the examples can improve the performance of some of the models, yet it does not have a positive effect on the models that already perform well.

9 Limitations and Future Research

This chapter starts with a discussion of the limitations of the thesis and potential solutions for future replications, and finishes with a proposal for a future direction for this line of research.

There are some limitations to this thesis that should be addressed in further research. First of all, limited time and resources prevented the tuning and annotation of larger amounts of data. Firstly, the models **SNLI$_t$**, **AR+SNLI$_t$**, **Ver+SNLI$_t$**, **Ver+AR+SNLI$_t$**, **Ver*+SNLI$_t$** and **Ver*+AR+SNLI$_t$** used only a subset of the *SNLI* data. There might be space for improvement if the full dataset was used, because larger datasets usually yield better results. The experiments should be replicated with the full dataset.

Secondly, only a small amount of examples from annual reports was annotated, as getting good annotations is a labor-intensive process that requires a well-instructed annotator. The disadvantage of the small size of the *AR* dataset is that the **AR** model learns only features that are very specific to this domain, which indicates that the model is probably not scalable. In addition, the effect of this dataset cannot be fairly compared with the larger datasets because data size can have a very large impact on the performance of neural models. A higher number of sentence pairs from the specific domain could be annotated for validating the comparisons in the future.

Thirdly, for limiting the amount of the *AR* data that needed to be annotated, target words had to be defined for selecting the premises. Only sentences that contain words such as 'emissions' and 'waste' were extracted from the reports. The need for defining these target words makes the thesis fall short of the proposition to solve the problems of specified event extraction, as the words denoting the participants of the event still have to be explicitly defined. This restriction on the premise sentences also has consequences for using them in training, as shown in Experiment 2. Nonetheless, if there were no time limitations, this problem could be avoided and all sentences from the reports could be used as premises. All the premises could be tested against the hypotheses that denote achievements on the sustainability front. That way any possible phrasings of relevant information could be covered, so that even sentences that don't explicitly mention the target words could be deduced to entail the hypothesis. In that case the results of all sentence pairs could be combined using a heuristic, as it has been done in other research that deals with entailment from multiple sources [Lai et al., 2017]. Following the strict definition of entailment, if at least one sentence entails an achievement of a target, the document could be classified as entailing it.

Fourthly, the method for recasting the event factuality data to the sentence pair format proves to have drawbacks. Namely, the resulting sentence pairs are too similar to each other as the hypothesis is always composed of the subset of the words in the premise. The attempt to remedy that by inserting random words into both sentences provides evidence that the rigid form of the sentences is indeed inhibiting the performance of the models trained on the *Ver* data,

however it is not enough to improve the best performing models. Another way to diversify the sentences could be substituting some words with their synonyms as well as dropping some less important words.

Finally, the logical definition of entailment had to be compromised in some cases, where a different semantic phenomenon than non-veridicality was responsible for the absence of entailment. For instance, strictly speaking one cannot determine that the premise entails the hypothesis in 30.

(30) PREMISE we has become the first chemical distributor to win the Lean & Green and Lean & Green Star awards under this program for demonstrating 20% CO_2 reduction in a 5-year period *in the Benelux and Italy.* [IMCD Annual Report, 2016]
HYPOTHESIS CO_2 was reduced.

Even though the CO_2 emissions were reduced in some areas, it is possible that the CO_2 emissions increased in another 14 countries and overall no reduction has been achieved.

However, in this thesis such sentence pairs were labeled with 'entailment'. This subtle issue is set aside for future research, because the models presented in this thesis target the notion of non-veridicality and are not equipped to deal with other fine-grained semantic distinctions. The absence of entailment between the two sentences in the example could be accounted for with the notion of monotonicity, which is discussed in the remaining part of this chapter.

Monotonicity (encompassing upward and downward entailment) is another semantic notion besides non-veridicality that is relevant for determining textual entailment. It has received a lot of attention in linguistics yet not in NLP research. The definitions of upward and downward entailment from [Giannakidou, 1999] can be found in Definition 4 and Definition 5.

Definition 4 *An operator* O *is downward entailing (DE) if and only if whenever* A *entails* B*,* O(B) *entails* O(A)*.*

Definition 5 *An operator* O *is upward entailing (UE) if and only if whenever* B *entails* A*,* O(B) *entails* O(A)*.*

One example of NLP work on this topic is a study by [Danescu-Niculescu-Mizil et al., 2009] based on the theoretical framework of [Ladusaw, 1980] who claims that downward entailing operators licence negative polarity items (NPIs) such as 'any'. They build an unsupervised algorithm that learns which operators are downward entailing ('deny', 'hardly', 'decline', etc.) based on the NPIs they co-occur with. The next step in the computational work in this area would be to construct systems that automatically learn the relation between sentences, by using the information about the DE and UE operators in them.

For the problem in this thesis, one would need to be able to detect non-UE operators, such as 'reduce' and 'stop'. The fact that premise a. does not entail the hypothesis in example 31 illustrates that 'stop' is not upward entailing.

(31) PREMISE a. He stopped smoking around his children.
PREMISE b. He stopped smoking on principle.
HYPOTHESIS He stopped smoking.

'Stop' is non-UE because while 'smoking around one's children' entails 'smoking', one can stop smoking around children without completely quitting. Equivalently, the lack of entailment between the sentences in example 30 above illustrates that 'reduce' is also not UE, because one can reduce CO_2 emissions in some locations without reducing them in absolute terms.

For textual entailment, aside from determining whether there is a non-UE operator, one would also have to determine whether there is a superset-subset relation between entities mentioned in a given sentence pair. The current textual entailment systems are able to detect when hyponyms and hypernyms are used in sentences when that is relevant for the entailment decision. However, sometimes hierarchical relations are also expressed with modification. It would be interesting to test whether the textual entailment systems are able to detect the relevance of modifiers. Determining what the modifier applies to in the sentence would be important for resolving whether the non-UE attribute is relevant. That is, both premises in example 31 contain the non-UE verb 'stopped', however premise a. does not entail the hypothesis while premise b. does. That is because 'around his children' restricts the meaning of 'smoking', whereas 'on principle' scopes over the non-UE operator 'stopped' and does not produce a proper subset of 'smoking'.

By introducing this phenomenon into the research on learning textual entailment, more precise relations could be determined. One could find out whether two proteins do not interact at all or only in particular circumstances, whether someone did not interfere with any presidential elections at all or only spared some, and other important distinctions. In the case of this research, it could help extract more precise information about the achievements of companies with regard to SDGs.

10 Conclusion

This research thesis presents one of the first explorations of not only what neural textual entailment models learn but also how they can be improved by fusing different datasets in training. Furthermore, this thesis suggests using the method of textual entailment for event extraction and factuality checking, applied specifically to the domain of reports on sustainability.

It appears that the method of combining datasets that contain different linguistic information for training a neural textual entailment model is a promising approach, as the model can learn to cover aspects of lexical and compositional semantics, such as veridicality. To recap, Experiment 1 shows that general linguistic inference data is not sufficient for testing data from a narrow domain even when tuning the model to the test domain. Experiment 2 shows that adding domain-specific data helps the entailment model to learn features that are idiosyncratic to the data, with the model relying on cues more than logical relations. Moreover, Experiment 3 shows that a textual entailment model trained on a combination of domain-specific and linguistically specialized data yields good results.

Furthermore, this thesis provides a framework of using RTE systems to address the tasks of event extraction and event factuality. Namely, a query sentence denoting an event can be tested for entailment against premises in a given text. This way events that are described with different wordings of the event participants and events with varying levels of factuality can be found. In the domain of company reports, one can find the accomplishments of companies with regard to reducing their environmental footprint even from densely hedged sentences.

Nevertheless, there remain numerous uncharted directions that should be researched in the future. First, some drawbacks of the current study could be corrected, such as expanding the research to include the whole *SNLI* dataset, annotating more *AR* data and adding more variation to the recast *Ver* dataset. Secondly, the research could be expanded to cover other semantic phenomena such as monotonicity.

By and large, this thesis corroborates the results of previous research which show that RTE models trained on SNLI data are biased. In addition, it shows that training the model on more diverse data can expand the coverage of the model with regard to the semantic phenomena present in the data. Last but not least, this thesis shows that interesting patters can be discovered in the linguistic representations built by the model, by analyzing the attention mechanism of the LSTM. Continuing the research into the inner workings of neural NLI models can help bridge the gap between textual entailment and logical entailment, training models that aim for the rigour of logical systems and the scalability of neural systems.

References

[Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[Bjerva et al., 2014] Bjerva, J., Bos, J., Van der Goot, R., and Nissim, M. (2014). The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

[Chen et al., 2016] Chen, Q., Zhu, X., Ling, Z., Wei, S., and Jiang, H. (2016). Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

[Ciampaglia et al., 2015] Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

[Dagan et al., 2006] Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

[Dagan et al., 2013] Dagan, I., Roth, D., Sammons, M., and Zanzotto, F. M. (2013). Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

[Danescu-Niculescu-Mizil et al., 2009] Danescu-Niculescu-Mizil, C., Lee, L., and Ducott, R. (2009). Without a'doubt'?: unsupervised discovery of downward-entailing operators. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 137–145. Association for Computational Linguistics.

[De Marneffe et al., 2014] De Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

[de Marneffe et al., 2011] de Marneffe, M.-C., Manning, C. D., and Potts, C. (2011). Veridicality and utterance understanding. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 430–437. IEEE.

[Giannakidou, 1999] Giannakidou, A. (1999). Affective dependencies. *Linguistics and Philosophy*, 22(4):367–421.

[Giannakidou, 2002] Giannakidou, A. (2002). Licensing and sensitivity in polarity items: from downward entailment to nonveridicality. *CLS*, 38:29–53.

[Giannakidou, 2006] Giannakidou, A. (2006). Only, emotive factive verbs, and the dual nature of polarity dependency. *Language*, 82(3):575–603.

[Glockner et al., 2018] Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

[Groenendijk, 1999] Groenendijk, J. (1999). The logic of interrogation: Classical version. In *Semantics and linguistic theory*, volume 9, pages 109–126.

[Harabagiu and Hickl, 2006] Harabagiu, S. and Hickl, A. (2006). Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912. Association for Computational Linguistics.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hunter, 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

[Hutchinson, 2004] Hutchinson, B. (2004). Acquiring the meaning of discourse markers. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 684. Association for Computational Linguistics.

[Jones et al., 2001] Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed <today>].

[Junfeng, 2016] Junfeng, H. (2016). Reasoning attention. https://github.com/junfenglx/reasoning_attention.

[Karttunen and Zaenen, 2005] Karttunen, L. and Zaenen, A. (2005). Veridicity. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[Kim et al., 2009] Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

[Kim et al., 2011] Kim, J.-D., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N., and Tsujii, J. (2011). Overview of bionlp shared task 2011. In *Proceedings of the BioNLP shared task 2011 workshop*, pages 1–6. Association for Computational Linguistics.

[Ladusaw, 1980] Ladusaw, W. A. (1980). Polarity sensitivity as inherent scope relations. *Garland Press, New York*.

[Lai et al., 2017] Lai, A., Bisk, Y., and Hockenmaier, J. (2017). Natural language inference from multiple premises. *arXiv preprint arXiv:1710.02925*.

[Le and Zuidema, 2015] Le, P. and Zuidema, W. (2015). Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*.

[Lee et al., 2015] Lee, K., Artzi, Y., Choi, Y., and Zettlemoyer, L. (2015). Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648.

[Manning et al., 2014] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

[Marcus et al., 1994] Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Marvin and Koehn, 2018] Marvin, R. and Koehn, P. (2018). Exploring word sense disambiguation abilities of neural machine translation systems. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas*, volume 1, pages 125–131.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Nédellec et al., 2013] Nédellec, C., Bossy, R., Kim, J.-D., Kim, J.-J., Ohta, T., Pyysalo, S., and Zweigenbaum, P. (2013). Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

[Özgür and Radev, 2009] Özgür, A. and Radev, D. R. (2009). Detecting speculations and their scopes in scientific text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1398–1407. Association for Computational Linguistics.

[Palangi et al., 2014] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. (2014). Semantic modelling with long-short-term memory for information retrieval. *arXiv preprint arXiv:1412.6629*.

[Penka and Zeijlstra, 2010] Penka, D. and Zeijlstra, H. (2010). Negation and polarity: an introduction. *Natural Language & Linguistic Theory*, 28(4):771–786.

[Poliak et al., 2018a] Poliak, A., Belinkov, Y., Glass, J., and Van Durme, B. (2018a). On the evaluation of semantic phenomena in neural machine translation using natural language inference. *arXiv preprint arXiv:1804.09779*.

[Poliak et al., 2018b] Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., and Van Durme, B. (2018b). Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81.

[Poliak et al., 2018c] Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018c). Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

[Rocktäschel et al., 2015] Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. (2015). Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

[Rudinger et al., 2018] Rudinger, R., White, A. S., and Van Durme, B. (2018). Neural models of factuality. *arXiv preprint arXiv:1804.02472*.

[Saurí and Pustejovsky, 2009] Saurí, R. and Pustejovsky, J. (2009). Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227.

[SDGs, 2015] SDGs, U. (2015). United nations sustainable development goals.

[Valencia et al., 1993] Valencia, V. S., Van der Wouden, T., and Zwarts, F. (1993). Polarity, veridicality, and temporal connectives. In *Proceedings of the 9th Amsterdam Colloquium, University of Amsterdam.*

[Vlachos and Riedel, 2014] Vlachos, A. and Riedel, S. (2014). Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22.

[White et al., 2016] White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., Rawlins, K., and Van Durme, B. (2016). Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

[Williams et al., 2017] Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

[Zaenen and Karttunen, 2013] Zaenen, A. and Karttunen, L. (2013). Veridicity annotation in the lexicon? a look at factive adjectives. In *Proceedings of the 9th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 51–58.

[Zhao et al., 2014] Zhao, J., Zhu, T., and Lan, M. (2014). Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 271–277.