# SYMBOLIC AND NEURAL APPROACHES TO NATURAL LANGUAGE INFERENCE

Hai Hu

Submitted to the faculty of the University Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

in the Department of Linguistics,

Indiana University

June 2021

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Lawrence S. Moss, PhD

Sandra Kübler, PhD

Chien-Jer Charles Lin, PhD

Donald Williamson, PhD

Date of Defense: 06/04/2021

Dedicated to my parents.

## ACKNOWLEDGEMENTS

This dissertation would not have been possible without the help and guidance I received from many people. I would like to express my sincere gratitude towards all of them.

First and foremost I would like to thank my co-advisors Larry Moss and Sandra Kübler. Larry has taught me how to think like a logician and has introduced me into the interesting and also challenging world of natural language inference. This dissertation would not have been started without his encouragement. I still remember the first time I was introduced to monotonicity in a talk given by Larry in our colloquium at Ballantine Hall. I had no idea that this interest would grow to an entire dissertation, as my initial career goal was to become a generative syntactician. Larry patiently guided me into the study on monotonicity, despite the fact that I knew next to nothing about logic at that time. It is our weekly meetings and intensive discussions that formed the core of the two symbolic systems in the first half of the dissertation. I owe him many of the ideas in the two chapters.

As a person who switched to computational linguistics only in the second year of the Ph.D. program, I can only write this dissertation because of what I learned from Sandra on how to conduct experiments scientifically and write clearly as a computational linguist. These skills have benefited my academic life tremendously and will continue to play an important role in my future career. Sandra meticulously went over all the chapters in great detail, without which the dissertation would have been in a much worse shape. She taught me how to be a good writer and how to adjust the writing to different audience and readership. Writing is an art to be improved for a lifetime, but I am glad to have laid a good foundation at IU.

My heartfelt thanks also go to the other two committee members: Chien-Jer Charles Lin and Donald Williamson, who have been more than supportive of the dissertation. Charles has also been a wonderful mentor in other projects we collaborate on. The deep learning course offered by Donald was the first formal introduction I had to neural modeling; I wish

I had taken it for credit rather than merely auditing.

Apart from the dissertation, my committee members have been the best mentors and collaborators I can hope for. I am very fortunate to have worked with them on several other projects, where their knowledge, expertise and professionalism have taught me how to be a better scholar.

My research life would also have been very unproductive if I did not have the opportunity to work with my many wonderful collaborators: Kyle Richardson, Qi Chen, He Zhou, Zuoyu Tian, Yiwen Zhang, Yina Ma, Aini Li, Jiahui Huang, Yanting Li, Wen Li, Liang Xu, Xuanwei Zhang, Daniel Dakota, Ken Steimel, Atreyee Mukherjee and Liang Zou. I have enjoyed the zoom meetings and the papers we wrote together would have been impossible without your hard work. I especially want to thank Kyle, Qi, He, Zuoyu, Yiwen, Yina and Yanting, for their help and work in some chapters of the dissertation.

Most of the work presented in the dissertation have been presented at a conference, and I would like to thank all the reviewers for their comments and suggestions.

I am also indebted to the supercomputer infrastructure at IU without which none of the experiments could be run.

I was able to have a happy social life and not quit the program in the past six years only because of the many friends I have in Bloomington and elsewhere: Donna Terry, Dennis Terry, Zhao Yang, Yayun Zhang, Russell Qin, Yiwen Zhang, Shiyue Sun, Danyao Li, Ruoze Huang, Jun Zhao, Qi Chen, Jian Liu, Yue Chen, Daniel Dakota, Ken Steimel, Atreyee Mukherjee, Zeeshan Ali Sayyed, Meng Li, Yining Wang, Peng Shen, Shuyu Guo, Junjie Guo, Xin Pang, Aolan Mi, Siying Ding, Jingyi Guo, Feier Gao, Xiao Dong, Yanting Li, Misato Hiraga, James C. Wamsley, Lan Yu, Amanda Foster, Jeyoung Park, Meng Wu, Xilin Li, Peiran Zhang, Yanan Feng, Meize Guo, Hao Yi, and finally anyone I might have forgotten to include in the list and those I have met at either the badminton court or different potlucks.

I also want to thank the professors I met at IU, either through collaboration or simply

in my six years of life in Bloomington. First I owe a big thank-you to Douglas Hofstadter for his (and his family's) generosity and kindness. I have always had a good time and tasty pizzas in our gatherings in Mother Bear's Pizza (zeugma intended). I unfortunately cannot make the title of my dissertation "Huh, AI?", which is what he suggested after realizing my username in IU's system is "huhai". I also want to thank Patrícia Amaral for giving me the chance of gaining a very deep understanding of one particular word in Spanish (*algo*). I am grateful to Vivek Astvansh for involving me in an interesting inter-disciplinary project where knowledge in computational linguistics seems to play a non-trivial role.

My appreciation also goes to my friends in China who have helped me in data collection, without whom one dissertation chapter would not have come into being: Ruoze Huang, Xiaojie Gong, Licen Liu and Jueyan Wu. Thanks to all the students who have contributed to the annotation of OCNLI.

Finally, I want to express my deepest gratitude to my family, who have supported and loved me unconditionally. Thanks to my girlfriend Aini Li who is not only a wonderful partner in life but also an incredibly patient listener to all my academic and non-academic "ramblings".

Hai Hu

SYMBOLIC AND NEURAL APPROACHES TO NATURAL LANGUAGE INFERENCE

Natural Language Inference (NLI) is the task of predicting whether a hypothesis text is entailed (or can be inferred) from a given premise. For example, given the premise that *two dogs are chasing a cat*, it follows that *some animals are moving*, but it does not follow that *every animal is sleeping*. Previous studies have proposed logic-based, symbolic models and neural network models to perform inference. However, in the symbolic tradition, relatively few systems are designed based on monotonicity and natural logic rules; in the neural network tradition, most work is focused exclusively on English.

Thus, the first part of the dissertation asks how far a symbolic inference system can go relying only on monotonicity and natural logic. I first designed and implemented a system that automatically annotates monotonicity information on input sentences. I then built a system that utilizes the monotonicity annotation, in combination with hand-crafted natural logic rules, to perform inference. Experimental results on two NLI datasets show that my system performs competitively to other logic-based models, with the unique feature of generating inferences as augmented data for neural-network models.

The second part of the dissertation asks how to collect NLI data that are challenging for neural models, and examines the cross-lingual transfer ability of state-of-the-art multilingual neural models, focusing on Chinese. I collected the first large-scale NLI corpus for Chinese, using a procedure that is superior to what has been done with English, along with four types of linguistically oriented probing datasets in Chinese. Results show the surprising transfer ability

of multilingual models, but overall, even the best neural models still struggle on Chinese NLI, exposing the weaknesses of these models.

---

Lawrence S. Moss, PhD

---

Sandra Kübler, PhD

---

Chien-Jer Charles Lin, PhD

---

Donald Williamson, PhD

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Natural language inference (NLI) is a fundamental task of natural language understanding (NLU) and an important goal of recent natural language processing (NLP) systems.

In an NLI problem, a text is given as the premise $\mathcal{P}$ for inference (sometimes also referred to as the "context"), and the goal of a human or a computer system is to decide whether a hypothesis $\mathcal{H}$ can be inferred from the premise, as shown in the following examples.

(1.1) $\mathcal{P}$ : *Every Asian linguistics student speaks at least 3 languages.*

$\mathcal{H}$ : *Every Chinese linguistics student speaks more than 2 languages.*

(1.2) $\mathcal{P}$ : *Some Asian linguistics student speaks at least 3 languages.*

$\mathcal{H}$ : *Some Chinese linguistics student speaks more than 2 languages.*

Examples like (1.1) and (1.2), are usually discussed within the field of logic and formal semantics, where "$\mathcal{P}$ entails $\mathcal{H}$" is typically defined as, in all possible worlds where $\mathcal{P}$ is true, $\mathcal{H}$ is also true. In example (1.1), to correctly conclude that $\mathcal{P}$ entails $\mathcal{H}$, a computer system needs to determine that *at least 3* entails *more than 2*, and that *every Asian linguistics student* contains *every Chinese linguistics student*. That is, in all possible worlds where *every Asian linguistics student* is capable of speaking 3 or more languages, so do every Chinese linguistics student because of the containment relation between *Asian* and *Chinese*. However, for another quantifier *some* in the same structure, shown in (1.2), the relation is non-entailment. That is, in a world where we believe $\mathcal{P}$ to be true, we cannot

commit ourselves to the truth condition of $\mathcal{H}$, simply because the Asian student mentioned in the $\mathcal{P}$ may not be Chinese. These two examples with different quantifiers are central to the *natural logic* tradition in the study of logic and formal semantics, which proposes and builds logical systems that resemble *natural* language for the task of inference (van Benthem, 1986; Icard and Moss, 2013; MacCartney, 2009; Sánchez-Valencia, 1991; Yanaka et al., 2019a).

NLI is also (if not more) interested in examples requiring some common sense and world knowledge that go beyond pure logical and semantic reasoning, as shown in example (1.3). For this entailment relation, a system has to understand that *forgetting to do something* usually means *not doing it*, a case of factive verbs well studied in the literature (De Marneffe et al., 2019; Nairn et al., 2006) and that *leaving something in the car trunk* entails *not bringing it to the house*, according to some general world knowledge about grocery shopping.

(1.3) $\mathcal{P}$ : *John forgot to take the milk out of his car trunk last night and he only discovered this today.*

$\mathcal{H}$ : *The milk was not brought into the house yesterday.*

Example (1.3) also illustrates the "informal" nature of many of the NLI problems. Under this "informal" assumption, "entailment" can be defined as whether an ordinary person is likely to infer $\mathcal{H}$ from $\mathcal{P}$ (Dagan et al., 2005; Christopher D Manning, 2006; Pavlick and Kwiatkowski, 2019), considering not only logical entailment, but also speaker intention, pragmatic implicature, world knowledge, etc. For instance, strictly speaking, it is possible that in (1.3) someone other than John brought the milk into the house yesterday, thus rendering $\mathcal{H}$ a non-entailment (although highly unlikely because John *only discovered this today*). However, as an ordinary person is unlikely to make such a judgment based on the

given $\mathcal{P}$ as well as our general knowledge about the situation, (1.3) is still considered a valid entailment in NLI.

Thus the term natural language inference can be said to cover two somewhat diverging definitions. One is the more strict logical and semantic entailment, while the other is the more loose and less formal inference that ordinary people make everyday. In fact, the task of NLI was named Recognizing Textual Entailment (RTE) at it inception (Dagan et al., 2005), but gradually researchers have adopted the term "inference" to encompass a wider range of phenomena. Therefore throughout the history of NLI research, there have been systems and datasets that target either the more strict definition of logical entailment, or the informal definition of inference.

In the logical and semantic tradition, many symbolic systems have been proposed. Typically, they translate the natural language input to some logical representation (for instance first-order logic or higher-order logic) and then call on theorem provers to find proofs that connect $\mathcal{P}$ and $\mathcal{H}$ (Abzianidze, 2016a; Bjerva et al., 2014; Martínez-Gómez et al., 2017; Yanaka et al., 2018). These symbolic systems can achieve very high precision, but the error-prone and challenging automatic translation of natural language into logical forms has caused problems (MacCartney and Christopher D Manning, 2008). The first half of the dissertation, chapters 3 and 4, therefore takes the *natural logic* approach, where the syntax of the logical system is based on natural language thus requiring no translation to any specific logical form. Building on the idea of monotonicity calculus, we have designed and implemented a symbolic inference engine that performs competitively with previous systems, while being more light-weight.

On the other hand, with the advent of neural network models, there has been growing interest from the machine learning community in NLI, mostly working under its "informal" definition. Models based on different neural architectures and configurations have been proposed (Q. Chen et al., 2017; Nie and Bansal, 2017). More recently, neural models such as the Bidirectional Encoder Representations from Transformers (BERT, Devlin et al.

(2019)) and XLNet (Z. Yang et al., 2019) have been introduced, which achieved impressive results on several NLI datasets.

This leads us to another key component in NLI research: the NLI datasets/corpora on which researchers train and evaluate their systems (the datasets are sometimes referred to as "benchmarks" in more recent literature). In this respect, one development over the past decade or so is the use of crowd-workers as annotators. Before 2010, most if not all the NLI datasets that the researchers have been working on are created by experts (for instance, the FraCas dataset, Cooper et al. (1996)). However, at the moment, most of the influential NLI datasets, just like other fields in artificial intelligence (AI), involve annotation from crowd workers (for instance, the Multi-genre NLI corpus, Williams et al. (2018)). Annotations from crowd-workers are assumed to represent the decisions of an ordinary person without formal training in linguistics and logic. Using crowd workers also means that it is easy to collect extremely large datasets. For instance, the Multi-genre NLI corpus collected a total of 443k problems ($\mathcal{P}$-$\mathcal{H}$ pairs), compared with a mere 346 problems in expert-created FraCaS.

The need for large datasets is driven in large part by the development of large neural network systems, which are especially data-hungry. Models like BERT (Devlin et al., 2019), with billions or more parameters, require huge amounts of raw text data, as well as human labeled data for training and fine-tuning. Because of the key role the training and evaluation datasets play in current NLU research, the question of how to create high-quality, bias-free training data and wide-coverage evaluation data has become a lively research field (Samuel R. Bowman and Dahl, 2021). Unfortunately, so far, most of the NLI research is focused on English, and there are no NLI datasets in Chinese. Thus in order to jump-start research on Chinese NLI, chapter 5 of this dissertation provides the first large-scale, high-quality NLI dataset in Chinese, with enhanced annotation procedure than previously proposed for English. This corpus also allows us to perform many interesting experiments with a variety of neural models.

Furthermore, there has been growing interest in training multilingual neural models (Conneau et al., 2020; Devlin et al., 2019; Lample and Conneau, 2019; Xue et al., 2020), which are single models capable of solving NLI problems in multiple languages. Despite their current success, there are still many unanswered questions as to when and how such cross-lingual transfer will work (training on English but evaluate on Chinese, for instance). Chapter 6 of the dissertation presents our first step to understanding these models, by constructing four types of challenging NLI datasets in Chinese which target specific reasoning skills, and evaluating and analyzing the models' performance on these datasets.

At a high level, inference plays a central role in human intelligence. Any truly intelligent computers should be able to make correct inferences given natural language input. At a lower level of different NLP tasks, researchers have argued that a system capable of natural language inference can serve as a generic module for other NLP tasks that involve elements of semantic inference (Dagan et al., 2005; Potts, 2021). Empirically, NLI data and models have been shown to be useful to various degrees in question answering (Angeli et al., 2016; J. Chen et al., 2021; Harabagiu and Hickl, 2006), information retrieval (Angeli et al., 2015), paraphrasing, summarization (Falke et al., 2019; H. Li et al., 2018; Pasunuru et al., 2017), and few-shot learning (S. Wang et al., 2021). We discuss the connection between NLI and other NLP tasks in chapter 1.2.1.

## 1.1 Overview of Natural Language Inference

In this section, we will present an overview of: 1) four formulations of NLI tasks, especially on the definition of the inference relations in chapter 1.1.1; 2) the common approaches and datasets that are used in the NLI literature in chapter 1.1.2.

### 1.1.1 Inference/Entailment Relations

There are several ways of defining entailment relations in the literature. Most tasks and datasets formulate NLI as an $n$-way classification task, with the binary and three-way clas-

sification being the most common. However, there is also a continuous and non-discrete view of entailment in more recent literature (T. Chen et al., 2020; Pavlick and Kwiatkowski, 2019). We will discuss each of them below.

**Binary classification.** The simplest classification is to classify sentence pairs into either entailment or non-entailment. This is used in the first three Recognizing Textual (RTE) Entailment Challenges (Dagan et al., 2005; Giampiccolo et al., 2007; Haim et al., 2006), the HELP and MED datasets which are investigating monotonicity-related inferences specifically (Yanaka et al., 2019a,b), the HANS dataset that studies the biases in existing NLI datasets (McCoy et al., 2019), among others.[1] This is the coarsest classification. The advantage of binary classification is that it avoids the sometimes difficult distinction between neutral and contradiction, as we will see below.

**Three-way classification.** This classification scheme has three labels: entailment, neutral and contradiction, and is used in many of the most widely used NLI resources: SICK (M. Marelli et al., 2014), SNLI (Samuel R Bowman et al., 2015), MNLI (Williams et al., 2018), ANLI (Nie et al., 2020a), among others. In the instructions for the crowd source annotation for SNLI, the three classes are interpreted as: given a premise, the hypothesis "is definitely true" (entailment), "might be true" (neutral), or "is definitely false" (contradiction), respectively. This allows the crowd workers to have a working definition of an entailment, which is also in the spirit of the informal nature of the NLI task. To be more specific, given that a premise $\mathcal{P}$ is true, an entailment label states that the hypothesis $\mathcal{H}$ can be automatically inferred (i.e., always true), whereas contradiction means that $\mathcal{H}$ is always false; finally, neutral means that there are situations where $\mathcal{P}$ and $\mathcal{H}$ are both true, and also situations where $\mathcal{P}$ is true while $\mathcal{H}$ is false.

It is the last point that sometimes causes problems in the real world. For instance, when

[1] See chapter 2.1 for details of the datasets.

we have

(1.4) $\mathcal{P}$ : *John is riding a bike* $\mathcal{H}$ : *John is playing the piano*

it is difficult to say definitively whether this is a contradictory pair or a neutral pair. Normally, it is extremely unlikely that someone is riding a bike and playing the piano at the same time; however, one can still construct a strange situation where John is doing both things at the same time.[2]

In a binary classification scenario, this will not cause any problems, because we can easily give the pair a non-entailment relation (which is essentially a meta-class for neutral and contradiction). In three-way classification, the current solution in most NLI datasets such as SNLI/MNLI/ANLI is to leave it for the annotators to decide, after which a majority vote is taken. For instance, if 3 out of 5 annotators label the pair as contradiction, it will then receive that label. However, taking the majority votes does not solve such boarderline cases. Rather, it ignores the annotations for the minority label(s). One solution, as experimented in Pavlick and Kwiatkowski (2019) is to keep all the labels from the annotators and ask the model to predict the distribution of the labels. They computed the correlation between the distribution of the labels and the confidence scores for the labels given by the model, showing that the two are in fact not correlated, leaving much room for improvement.

In this dissertation, all the datasets we used/created follow the three-way classification scheme unless stated otherwise.

**MacCartney's 7 basic relations.** Perhaps the most complex (discrete) inference relations are the 7 basic relations defined in MacCartney (2009). Chapter 5 of MacCartney (2009) first extends the entailment relations defined in Sánchez-Valencia, 1991 to 16 set relations. He then focused on 7 of the 16 relations, ignoring the other 9, which are commonplace

[2] See an image at https://inhabitat.com/pedaling-pianist-strikes-a-chord-with-his-homemade-piano-bike/.

| symbol | name | example | set theoretic definition |
|---|---|---|---|
| $x \quad y$ | equivalence | *couch* *sofa* | $x \quad y$ |
| $x$ € $y$ | forward entailment | *crow* € *bird* | $x$ € $y$ |
| $x$ • $y$ | reverse entailment | *Asian* • *Thai* | $x$ • $y$ |
| $x$ ^ $y$ | negation | *able* ^ *unable* | $x$ X $y$ H ^ $x$ Y $y$ $U$ |
| $x$ \| $y$ | alternation | *cat* \| *dog* | $x$ X $y$ H ^ $x$ Y $y$ $U$ |
| $x \smile y$ | cover | *animal* $\smile$ *non-ape* | $x$ X $y$ H ^ $x$ Y $y$ $U$ |
| $x \# y$ | independence | *hungry* # *hippo* | all other cases |

Table 1.1: 7 basic entailment relations in MacCartney (2009)

in logic but rare in natural language (cases where one set is empty or the entire universe). For instance, it is odd to discuss the entailment relation between the set of all dogs and the empty set. These 7 relations are summarized in Table 1.1, which is taken from MacCartney (2009, pp. 79).

In the 7 basic relations, equivalence and negation are self-explanatory from the examples in Table 1.1. The forward entailment relation (€) and backward entailment relation (•) denote the opposite entailment relations, where the former states that $x$ (strictly) entails $y$ and the latter states that $y$ (strictly) entails $x$. The alternation (|) and cover ($\smile$) relations are probably less common in NLI literature. Alternation is analogous to the "coordinating" terms or sister terms in knowledge bases such as WordNet (Miller, 1995), where the two terms share a common hypernym (*feline* and *canine* have the same hypernym: *carnivore*).[3] Note that the union of the two terms in an alternation relation will not cover the entire universe, as indicated in its set-theoretic definition in Table 1.1 (for instance, apart from *cat*s and *dog*s, there are other entities in the universe), while the two terms in a cover relation does cover the universe $U$ (animal $\smile$ non-ape: every entity in the universe is either an animal or a non-ape or both).

**Non-discrete/Probabilistic view.** Although not adopted in any major NLI benchmark so far, there have been attempts to frame the NLI task in a probabilistic setting (e.g., Pavlick

[3] https://ws4jdemo.appspot.com/ is a good tool for computing the relations between two words in WordNet.

and Kwiatkowski, 2019). In Pavlick and Kwiatkowski (2019), they use a -50 to 50 slide bar and ask the annotator to rate "how likely it is that $\mathcal{H}$ is true given that $\mathcal{P}$ is true". Their instructions are as follows:

> ... assume that the first sentence (S1) is true, describes a real scenario, or expresses an opinion. Using your best judgment, **indicate how likely it is that the second sentence (S2) is also true, describes the same scenario, or expresses the same opinion** ...

After collecting the "entailment likelihood" annotation from 50 annotators, they show that some pairs from the existing NLI datasets are intrinsically ambiguous: $\mathcal{P}$: *Paula swatted the fly*, $\mathcal{H}$: *The swatting happened in a forceful manner*. For such a pair, some annotators gave ratings around 0, which implies that they believe *swatting* does not necessarily need to be *forceful*. However, another group of annotators gave very positive ratings, indicating that *swatting* does entail a *forceful* manner. Clearly, having one discrete label (either *neutral* or *entailment*) does not reflect the entirety of human judgment. Pavlick and Kwiatkowski (2019) do not provide any measure for annotator agreement, but they show that when fitting a $k$-component Gaussian Mixture Model to the distribution of annotations, 20% of the sentence pairs have a nontrivial second component (weight ¡ 0.2). This can also be corroborated from the annotation results of MNLI, where only about 58% of the pairs receive a unanimous decision (the same label from 5 annotators), suggesting that disagreement is common in NLI annotation.

ChaosNLI (Nie et al., 2020b) and UncertainNLI (UNLI, T. Chen et al. (2020)) adopt a similar approach. T. Chen et al. (2020) also collected continuous "entailment likelihood" annotations and formulated NLI as a regression task. That is, instead of predicting a label, the model is trained to return a likelihood of entailment and the evaluation metrics include Pearson correlation and mean square error (MSE). Their results show that in terms of MSE, a BERT model can achieve near-human performance in a regression-formulated NLI task, similar to results on the same dataset with 3-way discrete labels.

| two-way | three-way | MacCartney | non-discrete |
|---|---|---|---|
| entailment | entailment | equivalence | likelihood |
| | | (strict) forward entailment | |
| non-entailment | neutral | (strict) backward entailment | |
| | | cover | |
| | | independence | |
| | contradiction | alternation | |
| | | negation | |

Table 1.2: Comparison of 4 schemes for entailment relations

**Comparison of schemes** In Table 1.2, we compare all four schemes of entailment relations. The two-way and three-way schemes are most commonly used in NLI/RTE datasets. Nevertheless, MacCartney's seven basic relations still provide us with a more fine-grained classification to consider. His equivalence relation can be roughly understood as paraphrase. The backward entailment, cover, and independence relations are all considered to be neutral in the three-way classification scheme. Finally, negation is unambiguously a contradiction, but alternation can be somewhat difficult. As mentioned before, are "riding a bike" and "playing the piano" in a strict alternation relation, as defined in Table 1.1? If they are, then the pairs in (1.4) is contradictory; if one assumes both can happen at the same time, then they are either in the cover or independence relation.

It is worth noting that there are several other schemes for entailment relations in the literature: Sánchez-Valencia (1991) uses a four-way classification (equivalence, entailment, reverse entailment, no-containment). Zhang et al. (2017) uses a 5-point Likert Scale for common sense inference with the prompt: *The following statement is ___ to be true during or shortly after the context of the initial sentence* (very likely, likely, plausible, technically possible, or impossible).

### 1.1.2 Common approaches and datasets

In this section, we will very briefly review the common approaches and datasets for NLI, and and also point out the issues in previous research that motivate this dissertation. We

will provide a more thorough literature review in chapter 2.

The task of natural language inference was first introduced as Recognizing Textual Entailment (RTE) (Dagan et al., 2005). As mentioned before, in formal semantics, strict "entailment" usually means that in all possible worlds, when the premise $\mathcal{P}$ is true, it follows that the hypothesis $\mathcal{H}$ is also true. For the NLP community, identifying strict logical entailment is generally not the main goal; rather, researchers are looking for judgments from non-experts (or, persons on the street), based on the $\mathcal{P}$, as well as the person's common sense and world knowledge. Thus, in the description of the first RTE shared-task (Dagan et al., 2005), the authors wrote:

> We say that T entails H if, typically, a human reading T would infer that H is most likely true. This somewhat informal definition is based on (and assumes) common human understanding of language as well as common background knowledge. (note: T is the premise $\mathcal{P}$)

The informal nature of the task has—to a certain degree—led to the adoption of a more general term of "inference", which is also the foundation of more recent large-scale, crowd-sourced datasets such as Stanford NLI corpus (Samuel R Bowman et al., 2015) and MultiNLI (Williams et al., 2018). The various versions of the RTE datasets are based on news text, where the premises are sentences in news articles, and the hypotheses are hand-written by experts. In the MultiNLI corpus (Williams et al., 2018), the premises are extracted from texts in 10 different genres.

However, this is not to say that logical reasoning is not important for NLI/RTE. In fact, logic/prover-based systems were a major type of system in the RTE shared-tasks, as well as as for the larger SICK dataset (M. Marelli et al., 2014). Such systems usually first translate the natural language input into some logical representation, and then call a theorem prover to find formal proofs from the premise to the hypothesis (Abzianidze, 2017; Yanaka et al., 2018). While they have achieved satisfactory performance on certain NLI datasets, the translation from natural language to logical forms is a bottleneck for the systems (MacCart-

ney and Christopher D Manning, 2008). Furthermore, although many logic-based systems acknowledge the critical role *natural logic* plays in the NLI tasks, only the `NatLog` system explicitly annotates the monotonicity information for each sentence (MacCartney, 2009). As we mentioned before, natural logic is concerned with logical systems whose syntax closely mimics the syntax of natural language. A major phenomenon in natural logic is monotonicity, which is concerned with upward and downward entailment (indicated by the arrows) in sentences like $\mathcal{S}$: *every dog*$^{\uparrow}$ *swims*$^{\uparrow}$. Upward entailment ($^{\uparrow}$) states that substituting *swim* with a more general expression (i.e., one with a larger extension) such as *move* produces an entailment: $\mathcal{S}$ entails *every dog moves*; for the word *dog* in a downward entailment environment ($^{\downarrow}$), we need to perform the opposite to obtain an entailment: $\mathcal{S}$ entails *every beagle swims*. The key in this type of entailment is the monotonicity arrow (also referred to as *polarity*): $^{\uparrow}$ and $^{\downarrow}$. However, the automatic annotation of monotonicity in `NatLog` is based on heuristics, rather than solid theories in monotonicity calculus, or (relatively) well studied algorithms in the monotonicity literature. The above points motivate the first research question of the dissertation. That is, can we build a more reliable automatic monotonicity annotator, based on sound theories in semantics and logic? A natural follow-up question is whether one can build an inference engine using only monotonicity and/or natural logic related inference rules, and how far such a system can go. These two questions are to be addressed in the first half of the dissertation.

Neural network based models—especially pre-trained Transformer encoder models—have gained momentum in recent years (Devlin et al., 2019; Lan et al., 2019; Liu et al., 2019). In these models, words are represented by contexualized word embeddings, and attention mechanisms (Bahdanau et al., 2014; Vaswani et al., 2017) are employed so that during training, the model will learn which words to "pay attention to" in order to solve the inference problem. Various pre-training techniques and objectives have been proposed in the literature, but in general, the models are trained on some general language modeling task (for instance, the Masked-Language-Modelling (MLM) task where the objective is to

predict randomly masked words) on (extremely) large text corpora. After pre-training, the model is expected to have learned some general semantic representation. It will then be fine-tuned to become an expert on specific down-stream tasks such as NLI. This learning scheme for the Transformer models has produced a remarkable boost in the performance on standard NLU benchmarks, which are evaluation platforms that combine multiple NLU tasks under one evaluation scheme, such as the English GLUE (A. Wang et al., 2018) and SuperGLUE benchmarks (A. Wang et al., 2019), as well as the Chinese CLUE benchmark (Xu et al., 2020), all of which are collections of different NLU tasks and include several NLI tasks.

However, a major issue in this line of work using neural-network models is that model fine-tuning requires a large annotated NLI dataset, which is only available in English. Furthermore, while the neural models often achieve very high accuracy on standard NLU tasks, they are still very easy to be fooled by humans under an adversarial setting (Nie et al., 2020a). There is thus a large gap between their high performance on the benchmarks and their actual language understanding ability when they are put to use "in the wild" or when tested on out-of-domain evaluation data (Ribeiro et al., 2020). For this reason, it is crucial to understand whether the high accuracy of the neural models is due to their strong language understanding ability, or their ability to exploit artifacts and biases in the dataset (McCoy et al., 2019). One way to answer this question is to carefully design NLI datasets that target specific language understanding abilities, and evaluate a fine-tuned model on them.

Additionally, several recent multilingual neural models have been shown to be successful in cross-lingual transfer (Conneau et al., 2020; Devlin et al., 2019; Goyal et al., 2021; Lample and Conneau, 2019; Xue et al., 2020). That is, these models are first pre-trained on a large multilingual corpus with a language modeling objective, and then fine-tuned in a supervised manner on an English downstream task (for instance sentiment analysis). After this, the models will be able to solve the same downstream task in other languages, for which they have not seen any human-labeled data. Such models have been demonstrated to

perform surprisingly well for various tasks and languages (Artetxe et al., 2020; Choi et al., 2021; Khashabi et al., 2020). However, we are still not fully aware of the strengths and limitations of these models, nor do we know much about their transfer ability from English to a typologically very different language—Chinese.

Against the backdrop of neural modeling, the second half of the dissertation first creates the first large-scale Chinese NLI dataset, along with four probing NLI corpora in Chinese, targeting various reasoning skills that are needed for NLI and language understanding in general. It then comprehensively examines a series of neural models, especially the pre-trained Transformer encoders, under both the classic supervised learning setting and the cross-lingual transfer learning setting, to understand their NLI ability and expose their potential weaknesses.

## 1.2 Natural Language Inference and Natural Language Understanding

Natural language inference is not only an essential task in natural language understanding, but is also closely connected to other tasks in NLU. Furthermore, because of its central role in language understanding, NLI has also been used for learning general meaning representations and probing the quality of the learned representations in neural network models. We will briefly discuss these points in the section, demonstrating how research on NLI can be helpful for building models that are capable of real language understanding.

### 1.2.1 Connection to other NLU tasks

**NLI and other NLU tasks** The ability to make reliable inferences plays an important role in several NLP tasks such as Question Answering (for instance Angeli et al., 2016; J. Chen et al., 2021; P. Clark et al., 2020; Harabagiu and Hickl, 2006; Trivedi et al., 2019) and Information Retrieval (for instance Angeli et al., 2015; Christopher D Manning, 2006). For example, to answer the multiple choice question (example taken from P. Clark et al., 2020):

*Which form of energy is produced when a rubber band vibrates?*

(1) chemical (2) light (3) electrical (4) sound

it is unlikely that in a knowledge base we have in store the exact sentence that contains the answer: "sound energy is produced when a rubber band vibrates". However, as long as the computer can *infer* this key sentence from some other similar sentences that are likely to appear in the knowledge base, for example, "sound is caused by vibrations", "vibrations produce sound energy", etc., then it can help the system make the right prediction. In real-world applications where usually no choices are present, NLI could still be useful if the question can be first transformed into a declarative sentence $S$ (*some form of energy is produced when a rubber band vibrates*) and then the task of finding an answer can be formulated as finding the sentence that entails $S$ (for instance, *sound is caused by vibrations of rubber bands*). Thus if a robust NLI system is available, it could serve as the foundation of a question answering system.

NLI is also related to other real-world applications. For instance, in the field of education the task of student answer scoring has been formulated as NLI/RTE and a shared-task has been held jointly on answer scoring and RTE in 2013 (Dzikovska et al., 2013). The goal of student answer scoring is to classify a student answer into one of several classes (correct, incorrect and contradictory, for instance), based on a given reference answer. Treating the reference answer as the premise and the student answer as the hypothesis and applying a RTE model to the answer pairs have produced good results on the task (Sung et al., 2019).

**NLI for learning sentence representations** Because of the fundamental role of NLI in language understanding, it has been used as a learning objective to "cram" some basic semantic information into a neural model. That is, since solving NLI requires multi-faceted reasoning ability, then a model pre-trained on an NLI objective should acquire some basic

reasoning ability which may serve as a general meaning representation for other meaning-related downstream tasks. For instance, Conneau et al. (2017) show that training a neural network model in a supervised fashion on the Stanford NLI corpus (Samuel R Bowman et al., 2015) will result in a high performing sentence embeddings model, evaluated on a suite of NLU tasks such as sentiment analysis, product review classification, subjectivity classification, among others. Specifically, they show that training a sentence encoder model in a supervised manner on natural language inference data results in better sentence representations than training an encoder in an unsupervised manner on much more data for much longer time (for instance the SkipThought model, Kiros et al. (2015)). Their sentence encoder, dubbed as "InferSent", received much attention from the community and has been used as a baseline sentence meaning representation for much subsequent work (Poliak et al., 2018; Williams et al., 2018). Other methods for learning sentence representation also found the usefulness of using NLI as a learning objective in a supervised setting for training the sentence representations, for example as demonstrated in Subramanian et al. (2018).

**NLI for transfer learning** Another use case for NLI is as an intermediate training task in multi-task learning scenario. The idea is that for a pre-trained language model like BERT, one can perform intermediate (supervised) training on NLI first, before finally fine-tuning the model on the target task (Phang et al., 2020, 2018; Pruksachatkun et al., 2020). This is referred to as *Supplementary Training on Intermediate Labeled data Tasks* (STILTs) (Phang et al., 2018). Through extensive experimentation, Pruksachatkun et al. (2020) found that tasks requiring higher level reasoning such as NLI and question answering yield the best results for transfer learning for other NLU task, for instance sentiment classification. The difference between STILTs and the methods described in the previous paragraph is that STILTs trains the model with NLI after it has been pretrained on other tasks (for instance masked-language modeling), while methods in the previous paragraph essentially use NLI as a pretraining objective.

### 1.2.2 Using NLI to probe neural models

With the advent of pre-trained Transformer encoder models such as BERT and its family members (Devlin et al., 2019; Lan et al., 2019; Liu et al., 2019), model performance on NLU benchmarks has dramatically increased and even surpassed estimates of human scores (Nangia and S. Bowman, 2019). However, for out-of-domain data and other real world applications, Transformer models still struggle, and numerous studies have shown that it is still relatively easy to construct examples where the models make wrong predictions (McCoy et al., 2019; Nie et al., 2020a). Now the major challenge is to understand how Transformer models work. More specifically, do the neural models really understand human language, or are they simply finding shortcuts or biases in the datasets to achieve high scores? In other words, how do we interpret the high performance of the models? Are they performing the kind of linguistic, logic, world-knowledge reasoning that humans are assumed to do when solving NLU problems, or are they simply good at finding a mapping from input strings to output labels that exploits the biases in the dataset to maximize its performance? A fruitful line of work has been done to answer this question (McCoy et al., 2019; Richardson et al., 2020; Tenney et al., 2019a,b; Yanaka et al., 2019a), where NLI is used as the task for probing the models.

The NLI task is ideal for exploring models' understanding ability for several reasons. First, there exists large-scale NLI datasets such as SNLI and MNLI that can be used to train a neural model to have some general language understanding ability. Second, it is easy to construct examples that target reasoning skills of a specific linguistic phenomenon in the NLI format. Third, the annotation procedure can be easily explained to non-experts, making it easier to collect data in large quantities using crowd-source platforms.

Therefore, the main theme of chapter 5 and chapter 6 is on the one hand to examine whether neural models that worked remarkably well on English NLI are able to achieve similar success on Chinese, but on the other to probe the neural models on various linguistic and logic reasoning skills in Chinese, as well as their cross-lingual transfer ability.

## 1.3 Main Research Questions

Based on the issues in current NLI research with the symbolic and neural approaches that are briefly reviewed in section 1.1.2, as well the importance of NLI in NLU research discussed in section 1.2, we ask the following major research questions in this dissertation:

1. How can we build a symbolic system that automatically annotates monotonicity information of input sentences? How does our system compare with a previous system, on the evaluation data we constructed? We extend the van Benthem algorithm (van Benthem, 1986) to cover a wider range of compositional rules in the Combinatory Categorial Grammar formalism for this task. We also add other rules needed for processing natural language input. (chapter 3)

2. Can we build a light-weight, symbolic inference engine that relies solely on monotonicity and natural logic? How can natural logic rules be incorporated in such a system? We build on the system from the previous chapter, and add a simple replacement operation for generating inferences, based on existing knowledge bases such the WordNet (Miller, 1995) and relations extracted from input text. (chapter 4)

3. How do we create a challenging NLI dataset in Chinese? Can our enhanced annotation procedure result in a dataset with higher-quality and more difficult NLI data for the neural models? We experiment with multiple neural models and compare their performance on the newly created NLI dataset. (chapter 5)

4. Can the multilingual Transformer models perform well under a zero-shot, cross-lingual transfer scenario on several Chinese NLI datasets? Specifically, can they be fine-tuned on English NLI data only and then achieve good performance on NLI involving uniquely Chinese linguistic phenomena? We also compare the performance of the models fine-tuned on machine-translated dataset and our high-quality NLI dataset created in the previous chapter, aiming to examine the effectiveness of high-

quality training data in an era where machine-translated datasets are easy to obtain and commonly used for training. (chapter 6)

## 1.4 Overview of the Dissertation

In chapter 2, we will review the relevant NLI datasets, as well as the symbolic and neural approaches from previous literature.

Then the dissertation is structured such that chapter 3 and chapter 4 are on symbolic models, whereas chapter 5 and chapter 6 examine the neural approach.

Specifically, in chapter 3 and chapter 4, we propose a new symbolic model for NLI that is based on monotonicity and natural logic. Chapter 3 describes the `ccg2mono` system that automatically annotates monotonicity information on input sentences. `ccg2mono` extends the algorithm in van Benthem (1986) to cover more composition rules in the CCG formalism, and then relying on the CCG parse tree, it tags every constituent with monotonicity information which will be used to make inferences. We evaluate `ccg2mono` on a small expert-crafted evaluation dataset and show that it outperforms the `NatLog` system. Chapter 4 builds on chapter 3 and proposes an inference engine called `MonaLog` that generates inferences based on the monotonicity information provided by `ccg2mono`. We describe how the inference engine works and the choice of the knowledge base for inference generation, and then evaluate it on two commonly used datasets for symbolic and logic-based models. The results show that `MonaLog` performs on-par with previous models despite being relatively light-weight. We then discuss the challenges for symbolic modeling.

Chapter 5 and chapter 6 examine the neural models for NLI using crowd-sourced, large-scale NLI corpora. In particular, chapter 5 presents the first large-scale NLI corpus for Chinese: Original Chinese NLI (OCNLI) corpus, with a total of 56,000 examples, annotated by students with expertise in language. Experimental analyses show that the enhanced procedures in OCNLI made it a challenging benchmark, with the best model lagging 12% behind humans' performance. Further experiments on different subsets of OCNLI show

that while being more challenging than its English counterpart MNLI, OCNLI still contains hypothesis-only biases, which shows the challenges in better data collection. In chapter 6, we design four categories of adversarial and probing NLI datasets in Chinese, parallel to established literature in English, and then examine the cross-lingual transfer ability of a multilingual neural model—XLM-RoBERTa. Extensive experimentation suggests that the multilingual neural model fine-tuned on English data alone can outperform monolingual Chinese models when tested on several of our constructed Chinese evaluation data, even in 3 out of 5 linguistic phenomena that are unique in Chinese: idioms, pro-drop and non-core arguments. Mixing the expert-annotated OCNLI with existing English NLI data given further increase in the performance. Finally, we discuss the implications of the results on (cross-lingual) neural model probing and NLI dataset creation.

Chapter 7 concludes the dissertation by summarizing the main findings and contributions, and new research questions raised in the dissertation.

# CHAPTER 2

# DATASETS FOR NLI AND PREVIOUS APPROACHES

In this chapter, we will review previous datasets and approaches for NLI. I will go into details of the datasets used in this dissertation: FraCaS and SICK, and briefly introduce the other ones. With respect to computational approaches to NLI, we will focus broadly on logic-based and neural-network based methods that are related or used in this dissertation.

## 2.1 Natural Language Inference Datasets

Having high-quality training and evaluation datasets is key for any NLP task. Datasets in NLI/RTE have grown considerably in the past few decades, both in terms of dataset size as well as diversity. The first generation of datasets such as FraCaS (Cooper et al., 1996) and RTE (Dagan et al., 2005) are relatively small, with a few hundred or at most 1,000+ examples, which are suitable for symbolic and rule-based systems, but unsuitable for machine-learning and deep-learning models as they usually require more training examples. The SICK dataset (M. Marelli et al., 2014), with roughly 10,000 examples is the first resource that machine learning and deep learning models have been widely tested on. Since then, we have witnessed rapid creation of (large-scale) resources for general-purpose NLI.

### 2.1.1 NLI Datasets: General-purpose, Probing, and Adversarial

There are many ways of classifying existing NLI resources. One way is to classify them with respect to the purpose of the corpus, by which we will consider three classes. The first is **general-purpose NLI**, which are aimed at training/testing models for general inferences, rather than for specific reasoning abilities.

**General-purpose datasets** In this class, we have resources such as the Stanford NLI corpus (Samuel R Bowman et al., 2015) and Multi-genre NLI (Williams et al., 2018), both of which are created by providing pre-defined premises to an annotator and asking him/her to write hypotheses that conform to one of the three inference relations. We introduce them in chapter 2.1.3.

**Probing/diagnostic datasets** With the rise of NLI resources for general inference purposes, there is second class resources that examine specific reasoning abilities, which we call **probing or diagnostic** NLI dataset, for instance, the HELP (Yanaka et al., 2019b) and MED (Yanaka et al., 2019a) datasets for monotonicity reasoning, ConjNLI for conjunctive sentences (Saha et al., 2020), AddOne for inference of simple Adj+N phrases (Pavlick and Callison-Burch, 2016), Vashishtha et al. (2020) for temporal reasoning, semantic fragments (Richardson et al., 2020) and the GLUE diagnostics (A. Wang et al., 2018) for a collection of linguistic and logical reasoning abilities, among others.

We describe the semantic fragments (Richardson et al., 2020) in detail because they are used as basis for data creation in chapter 6. The goal of Richardson et al. (2020) is to examine whether neural models are capable of solving NLI problems targeting 7 linguistic and logic phenomena: negation, boolean coordination, quantification, counting, conditionals, comparatives, and monotonicity reasoning, which they named *semantic fragments*. In order to do so, the authors generated synthesized NLI examples based using context-free grammars and a vocabulary of countries, person names and animals, as shown in Table 2.1. The first 6 fragments are modified from examples in Salvatore et al. (2019), while the monotonicity examples are generated using the `MonaLog` system (which will be introduced in chapter 4). Experimental results with several neural models show that if only trained on general-purpose NLI corpora (SNLI/MNLI), the models have low performance on the semantic fragments (42–62% in accuracy). However, continued training on a few hundred examples from the in-domain data in the semantic fragments, the models

| Fragments | Example (premise,label,hypothesis) | Genre | Vocab. Size | # Pairs | Avg. Sen. Len. |
|---|---|---|---|---|---|
| Negation | *Laurie has only visited Nephi, Marion has only visited Calistoga.* CONTRADICTION *Laurie didn't visit Calistoga* | Countries/Travel | 3,581 | 5,000 | 20.8 |
| Boolean | *Travis, Arthur, Henry and Dan have only visited Georgia* ENTAILMENT *Dan didn't visit Rwanda* | Countries/Travel | 4,172 | 5,000 | 10.9 |
| Quantifier | *Everyone has visited every place* NEUTRAL *Virgil didn't visit Barry* | Countries/Travel | 3,414 | 5,000 | 9.6 |
| Counting | *Nellie has visited Carrie, Billie, John, Mike, Thomas, Mark, .., and Arthur.* ENTAILMENT *Nellie has visited more than 10 people.* | Countries/Travel | 3,879 | 5,000 | 14.0 |
| Conditionals | *Francisco has visited Potsdam and if Francisco has visited Potsdam then Tyrone has visited Pampa* ENTAILMENT *Tyrone has visited Pampa.* | Countries/Travel | 4,123 | 5,000 | 15.6 |
| Comparatives | *John is taller than Gordon and Erik..., and Mitchell is as tall as John* NEUTRAL *Erik is taller than Gordon.* | People/Height | 1,315 | 5,000 | 19.9 |
| Monotonicity | *All black mammals saw exactly 5 stallions who danced* ENTAILMENT *A brown or black poodle saw exactly 5 stallions who danced* | Animals | 119 | 10,000 | 9.38 |

Table 2.1: 7 semantic fragments proposed in Richardson et al. (2020), where the top four fragments test basic logic (`Logic Fragments`) and the last fragment covers monotonicity reasoning (`Monotonicity Fragment`). Examples taken from the original paper.

can quickly master the fragments, reaching near perfect performance on most fragments, except the comparative fragment.

**Adversarial/stress-testing datasets** Third, there is a series of **adversarial or stress-testing** datasets, created with the intention to test the limits of the models. That is, they try to construct difficult examples or examples that take advantage of the weaknesses of the models, in order to "break" these models, and also study how they perform when tested with data outside their domain. These include: the BreakingNLI corpus (Glockner et al., 2018), the English NLI stress-test (Naik et al., 2018), unnatural NLI (Sinha et al., 2020), position-related NLI (Y.-C. Lin and Su, 2021), and also the Adversarial NLI (Nie et al., 2020a), among others. This leads to re-examination of not only the limits of the model architecture and the procedure of NLI data creation, but also the sometimes over-stated claims about the understanding ability of state-of-the-art neural models.

We further detail the stress tests in Naik et al. (2018)[1] since some of our newly constructed datasets in chapter 6 are inspired by them. Specifically, based on an error analysis of the neural models on the MNLI dataset, they devised several stress tests to test the limits of the best neural models back then (Q. Chen et al., 2017; Nie and Bansal, 2017), as illustrated in Table 2.2. The categories of Antonyms, Word Overlap, Negation, Length Mis-

[1] https://abhilasharavichander.github.io/NLI_StressTest/

match and Spelling Errors used premise-hypothesis pairs from the MNLI corpus and either alter or add words to the hypothesis. The Antonyms condition randomly swap a word in the premise with its antonym to form a contradiction. In Word Overlap, Negation and Length Mismatch, one or more tautology (*true is true* or *false is not true*) are added to distract the model (but the inference label should not change). In Spelling Errors, they randomly switch two letters in a word. For numerical reasoning, they sampled sentences from math problems as premises and the generated the hypothesis by finding another expression for the number (e.g., *350* entails *less than 750*). Their results show that the best model back then (Nie and Bansal, 2017) trained on MNLI has a large discrepancy in performance on the in-domain MNLI dev set and the stress tests they constructed, which can be around 50 percentage points. That is, a well-performing model on the in-domain evaluation can fail catastrophically on the stress tests, suggesting the need for more building robust neural models. However, it is worth pointing out that the neural models they tested are much smaller and less powerful than the more recent transformer encoders such as BERT (Devlin et al., 2019); thus their results may not hold for the neural models we are testing in this dissertation.[2] In chapter 6, we build a Chinese stress tests following Naik et al. (2018) with several modifications, which we detail in chapter 6.3.1.

Another way of viewing the plethora of datasets is grouping them according to how they are created, i.e., whether they are annotated by experts, or crowd-sourced. I will review several datasets in detail under this classification because chapter 5 and chapter 6 are concerned with the procedure of data collection. We will only review the datasets that are used in this dissertation or have inspired the work reported in the dissertation.

### 2.1.2 Expert-created Datasets: FraCaS and GLUE/CLUE Diagnostics

Expert-created datasets usually require careful selection of the linguistic and logical phenomena to be included and also considerable amount of investment in time from the ex-

[2] See chapter C.4 for results of the more recent models.

| Category | Premise | Hypothesis | Label |
|---|---|---|---|
| **Antonyms** | I love the Cinderella story. | I hate the Cinderella story. | C |
| **Numerical Reasoning** | Tim has 350 pounds of cement in 100, 50, and 25 pound bags | Tim has less than 750 pounds of cement in 100, 50, and 25 pound bags | E |
| **Word Overlap** | Possibly no other country has had such a turbulent history. | The country's history has been turbulent and true is true | E |
| **Negation** | Possibly no other country has had such a turbulent history. | The country's history has been turbulent and false is not true | E |
| **Length Mismatch** | Possibly no other country has had such a turbulent history and true is true and true is true and true is true and true is true and true is true | The country's history has been turbulent. | E |
| **Spelling Errors** | As he emerged, Boris remarked, glancing up at teh clock: "You are early | Boris had just arrived at the rendezvous when he appeared | N |

Table 2.2: Examples from the stress tests in Naik et al. (2018), taken from the original paper.

perts. Thus they are usually much smaller than the crowd-sourced corpora, which are easy to collect on crowd-sourcing platforms.

**FraCaS** The FraCaS corpus[3] (Cooper et al., 1996) consists of 346 NLI questions, covering 9 broad logic and semantic phenomena: generalized quantifiers; plurals; (nominal) anaphora; ellipsis; adjectives; comparatives; temporal reference; verbs; attitudes. The problems in FraCaS are designed to have one to five premises. See examples in Table 2.3. As this dataset is extremely small, it is mostly used to evaluate symbolic systems (Abzianidze, 2016b; Angeli and C. Manning, 2014; Dong et al., 2014; M. Lewis and Steedman, 2013; MacCartney and Christopher D Manning, 2008; Mineshima et al., 2015; Tian et al., 2014). The categorization of 9 reasoning types lays out the kinds of semantic and logical phenomena a capable system needs to handle. In chapter 4.4, we propose a symbolic inference system based on theories of monotoncicity, and report an experimental evaluation on the first section (monotonicity) of FraCaS.

[3] https://nlp.stanford.edu/~wcmac/downloads/fracas.xml

| | |
|---|---|
| Example 19 | |
| Premise 1 | All Europeans have the right to live in Europe. |
| Premise 2 | Every European is a person. |
| Premise 3 | Every person who has the right to live in Europe can travel freely within Europe. |
| Question | Can all Europeans travel freely within Europe? |
| Hypothesis | All Europeans can travel freely within Europe. |
| Answer | yes |
| Example 337 | |
| Premise 1 | ITEL tried to win the contract in 1992. |
| Question | Did ITEL win the contract in 1992? |
| Hypothesis | ITEL won the contract in 1992. |
| Answer | unknown |

Table 2.3: Example 19 (monotonicity) and example 337 (intentional attitudes) from FraCaS

**GLUE/CLUE diagnostics** The General Language Understanding Evaluation benchmark is a collection of 9 NLU tasks (including acceptability judgment, sentiment analysis, natural language inference, etc.) where models can be evaluated and compared (A. Wang et al., 2018).[4] The GLUE diagnostics are designed to provide qualitative analysis of specific linguistic/logic phenomena for the submitted models, in addition to the average score of the models on the 9 tasks in GLUE. It includes roughly 1000 NLI examples that are designed to analyze the performance of a system on "a broad range of linguistic phenomena".[5] Unlike the FraCaS, which categorizes each problem into one of the 9 reasoning types, examples in GLUE diagnostics may fit into one or more pre-defined linguistic phenomena. The linguistic phenomena are defined at two levels, one coarse and one fine-grained. The coarse level has four categories: (1) Lexical Semantics; (2) Predicate-Argument Structure; (3) Logic; (4) Knowledge and Common Sense.

Each of the coarse level category has subcategories. For instance, under Lexical Semantics, there are a total of 7 subcategories: Lexical Entailmlent, Morphological Negation, Factivity, etc. There are another 8 subcategories for Predicate-Argument Structure, 4 for

[4] https://gluebenchmark.com/
[5] https://gluebenchmark.com/diagnostics

Logic and 2 for Knowledge and Common Sense. The GLUE diagnostics is used to perform a qualitative analysis of the models submitted to the GLUE benchmark.

Similarly, the Chinese Language Understanding Evaluation (CLUE) benchmark also includes a diagnostic dataset in the form of NLI (Xu et al., 2020). The CLUE diagnostics feature examples on 9 linguistic categories including anaphora, double negation, monotonicity, etc., with a total of 514 NLI pairs (see Table 2.4). In chapter 5, we compare models trained on our newly collected Chinese NLI dataset with those trained on the machine-translated XNLI dataset on the CLUE diagnostics. In chapter 6, we further expand the CLUE diagnostics by adding examples from 5 new linguistic categories (pro-drop, idioms, non-core argument, etc.) and doubling the examples in the 9 original categories. The expanded CLUE diagnostics, with a total of 2,121 NLI pairs, will be explained in chapter 6.3.2, and will be used to evaluate the cross-lingual transfer ability of the neural models.

### 2.1.3 Crowd-sourced Datasets: SICK and MNLI

**Sentences Involving Compositional Knowledge (SICK)** SICK is the first large-scale NLI dataset, with roughly 10k NLI pairs that can be used to train a neural model (M. Marelli et al., 2014). The premises are taken from image captions and the hypotheses are transformed from these captions via hand-crafted templates. Then each premise-hypothesis pair is labeled either "entailment", "neutral" or "contradictory" by five crowd-workers, where the majority vote is used as the gold label. Example NLI pairs from SICK are presented in Table 2.5. One issue of the SICK dataset is the unreliability (or the inconsistency) of its labels, as noted in Kalouli et al. (2017a, 2018), and also illustrated in Table 2.5. For instance, in problem 294, the two girls cannot be lying the sitting on the ground at the same time, and thus the label should be contradiction. For problem 1645, the premise seems nonsensical because it is hard to tell what "a jumping car" is.

Kalouli and her colleagues have been correcting the SICK annotations and have pro-

| | # | Premise | Hypothesis | Label |
|---|---|---|---|---|
| Anaphora | 48 | l 丽 Œ y „ Í 亲 N 4 一 w O ( Ù Ì 。<br>Ma Li and her mother Li Qin live here together. | l 丽 / N 4 „ Í 亲。<br>Ma Li is Li Qin's mother. | C |
| Argument structure | 50 | } Á ¢ ( S 8 。<br>Xiao Bai saw Xiao Hong playing video games. | ¢ ( S * • ó 。<br>Xiao Hong is doing Tai Chi. | C |
| Common sense | 50 | ¡ å \ 。<br>Xiaoming doesn't have a job. | ¡ O ? 。<br>Xiaoming doesn't have a place to live. | N |
| Comparative | 50 | Ù P T P Ô £ P 。<br>This basket has more oranges than that one. | Ù P T P Ô £ P 了不 。<br>This basket has much more oranges than that one. | N |
| Double negation | 24 | ` + 不 Š Å Û S 一 Þ 事。<br>Don't take minor illness as nothing. | ` ″ å Í Æ Å Û 。<br>You should pay attention to minor illness. | E |
| Lexical semantics | 100 | ¢ ˆ ¾ Ç 。<br>Xiaohong is sad. | ¢ ˆ ¾ 。<br>Xiaohong is ugly. | N |
| Monotonicity | 60 | 些 f œ " ( l q ¡ Ì 1 L 。<br>Some students like to sing in the shower room. | 些 s œ " ( l q ¡ Ì 1 L 。<br>Some female students like to sing in the shower room. | N |
| Negation | 78 | s ¿ 7 ÿ e 。<br>Girls dormitory, no entering for boys. | s ¿ ê ý s Û ú 。<br>Only girls can go in and out of the girls dormitory. | E |
| Time of event | 54 | ° » t Ç ¿ 业 ¶ 了。<br>The reporter interviewed the entrepreneur last year. | ° Ï 8 Ç ¿ 业 ¶ 。<br>The reporter interviews the entrepreneur very often. | N |

Table 2.4: Examples from the original CLUE diagnostics in Xu et al. (2020), with a total of 514 NLI pairs in 9 linguistic categories.

duced a partially corrected SICK (Kalouli et al., 2017b, 2018). They first manually checked 1,513 NLI pairs tagged as "A entails B but B is neutral to A" (*AeBBnA*) in the original SICK[6], correcting 178 pairs that they considered to be wrong (Kalouli et al., 2017b). Later, Kalouli et al. (2018) extracted pairs from SICK whose premise and hypothesis differ in only one word, and created a simple rule-based system that used WordNet information to solve the problem. Their WordNet-based method was able to solve 1,651 problems, whose original labels in SICK were then manually checked and corrected against their system's output. They concluded that 336 problems are wrongly labeled in the original SICK. Combining the above two corrected subsets of SICK, minus the overlap, results in their corrected SICK

[6]Note that in SICK the entailment relations are annotated both ways. That is, sentence 1 Ñ sentence 2 and sentence 2 Ñ sentence 1.

| id | premise | hypothesis | orig. label | corr. label |
|---|---|---|---|---|
| 219 | There is no girl in white dancing | A girl in white is dancing | C | C |
| 294 | Two girls are lying on the ground | Two girls are sitting on the ground | N | C |
| 743 | A couple who have just got married are walking down the isle | The bride and the groom are leaving after the wedding | E | N |
| 1645 | A girl is on a jumping car | One girl is jumping on the car | E | N |
| 1981 | A truck is quickly going down a hill | A truck is quickly going up a hill | N | C |
| 8399 | A man is playing guitar next to a drummer | A guitar is being played by a man next to a drummer | E | n.a. |

Table 2.5: Examples from SICK (M. Marelli et al., 2014) and corrected SICK (Kalouli et al., 2017b, 2018). n.a.: example not checked by Kalouli and her colleagues. C: contradiction; E: entailment; N: neutral.

| total | N Ñ E | E Ñ C | N Ñ C | E Ñ N |
|---|---|---|---|---|
| 409 | 14 | 7 | 190 | 198 |

Table 2.6: Changes from SICK to corrected SICK (Kalouli et al., 2017b, 2018).

dataset[7], which has 3,016 problems (3/10 of the full SICK), with 409 labels different from the original SICK (see breakdown in Table 2.6). 16 of the corrections are in the trial set, 197 of them in the training set and 196 in the test set. This suggests that more than one out of ten problems in SICK are potentially problematic.

At the moment no fully checked SICK is available and thus in chapter 4 we evaluate on the original and partially checked SICK.

SICK was used in the SemEval-2014 shared-task (Marco Marelli et al., 2014), and has received continuous attention from the community. It also inspired the much larger Stanford NLI (SNLI) corpus (Samuel R Bowman et al., 2015) which also uses image captions as premises. One difference is that SNLI hired annotators to produce hypotheses based on a given premise, rather than transforming the premises via templates into hypotheses, as has been done in SICK. At 570k, SNLI is more than 50 times larger than SICK, inspiring a long line of work in neural modeling of NLI.

[7] https://github.com/kkalouli/SICK-processing

**Multi-genre Natural Language Inference (MNLI)** The release of SNLI has sparked continuous interests in NLI, and by 2019 the performance on the test set has saturated and stayed around 90%.[8] Furthermore, using image captions as the sole source of premises excludes inferences on important phenomena such as temporal reasoning, belief and modality (Williams et al., 2018).

Thus to create a more diverse, challenging and balanced NLI corpus, Williams et al. (2018) sampled premises from 10 genres of text and followed the same two-step annotation procedure in SNLI to create the Multi-genre NLI (MNLI) corpus. Specifically, in step one, crowd workers were instructed to write three hypotheses (entailment, neutral and contradiction) according to a given premise; in step two, another four crowd workers were asked to verify the pair by giving it one of the three labels. Note that not all 433k pairs in the MNLI corpus have been verified, but all the pairs in the development and test sets are. See Table 2.7 for examples from MNLI.

With the more diverse premises and linguistic phenomena, the MNLI corpus turns out to be more difficult than SNLI and is included in the GLUE benchmark as one of its 9 NLU tasks (A. Wang et al., 2018).

### 2.1.4 Issues and Biases in NLI Resources

While resources such as SNLI and MNLI have allowed researchers to build and test different kinds of neural models and greatly accelerated the research on NLI, there is a line of work that identified biases and problems in these datasets.

Apart from the disagreement in annotation mentioned above (for SICK), there are two other very pronounced issues in widely used NLI datasets (such as SNLI and MNLI), which have received much attention in the literature.

First, the datasets are **not reflecting the genuine and diverse challenges in NLU**. In other words, they are too easy for the neural models and do not provide satisfactory evalu-

[8] See the progression of performance on the leaderboard: https://nlp.stanford.edu/projects/snli/.

| premise | GENRE<br>label | hypothesis |
|---|---|---|
| Met my first girlfriend that way. | FACE-TO-FACE<br>**contradiction**<br>C C N C | we didn't meet my first girlfriend until later. |
| 8 million in relief in the form of emergency housing. | GOVERNMENT<br>**neutral**<br>N N N N | The 8 million dollars for emergency housing was still not enough to solve the problem. |
| Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community. | LETTERS<br>**neutral**<br>N N N N | All of the children love working in their gardens. |
| At 8:34, the Boston Center controller received a third transmission from American 11 | 9/11<br>**entailment**<br>E E E E | The Boston Center controller got a third transmission from American 11. |
| I am a lacto-vegetarian. | SLATE<br>**neutral**<br>N N E N | I enjoy eating cheese too much to abstain from dairy. |
| someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny | TELEPHONE<br>**contradiction**<br>C C C C | No one noticed and it wasn't funny at all. |

Table 2.7: Examples of MNLI, copied from Williams et al. (2018), shown with their genre labels (FACE-TO-FACE, GOVERNMENT, etc.), their selected gold labels, and the validation labels (abbreviated E, N, C) assigned by individual annotators.

ation of these models. This is a problem for recent NLU datasets and benchmarks in general (Samuel R. Bowman and Dahl, 2021; Kiela et al., 2021). As probably the single most important way to measure progress in NLP, the evaluation data in these datasets are used to run different models on and then their results are compared. Improvements on the evaluation data, regardless of the metric being used (accuracy, F1, error rate, etc.), are generally acknowledged as improvements in the modelling. The issue in current NLP especially NLU research is that models that achieve high performance or even "super-human" performance (however that is defined) still struggle in real-world applications, and the general consensus among NLP practitioners is that these models are still far from having human-like language understanding abilities despite their high performance on the benchmarks (Samuel R. Bowman and Dahl, 2021; Kiela et al., 2021; Ribeiro et al., 2020). Thus we are in dire need of

better benchmarks (evaluation datasets or methods) that truly reflect the capabilities of the models and provide more realistic estimates of models' performance in real-world applications. The first step for fixing this problem would be to create datasets that include more difficult examples for current state-of-the-art models.

One recent attempt at fixing this issue is the Adversarial NLI (ANLI), Nie et al. (2020a) corpus, which uses the Human-And-Model-in-the-Loop Enabled Training method for data collection. Specifically, they first fine-tune a neural model on a combination of existing English NLI datasets, and then ask annotators to write examples to fool the model; that is, they are explicitly instructed to write NLI examples for which the model will make wrong predictions for at most 10 tries, a strategy commonly referred to as adversarial attack (Ettinger et al., 2017; Jia and Liang, 2017; Zellers et al., 2018). They show that this annotation strategy produces much harder NLI examples, with the state-of-the-art neural models performing at only about 50% accuracy. However, their annotation method requires an existing NLI corpus to train the model during annotation, which is not possible for Chinese at the moment, as there exists no high-quality Chinese data.

In chapter 5, we take another approach to make our Chinese NLI corpus more challenging, without using a HAMLET annotation method. Our method is detailed in chapter 5.3.2.

The second issue for SNLI and MNLI is that they **contain biases** that the models can exploit to achieve high scores without really learning to perform NLI (Geva et al., 2019; Gururangan et al., 2018; McCoy et al., 2019; Poliak et al., 2018; Tsuchiya, 2018).

Concretely, Gururangan et al. (2018), Poliak et al. (2018) and Tsuchiya (2018) discovered that a model can achieve high accuracy by only looking at the hypothesis and ignoring the premise completely (see also Feng et al. (2019)), which is usually referred to as hypothesis-only bias in the literature. These biases have been mainly associated with the annotators (crowd workers in MNLI's case) who use certain strategies to form hypotheses of a specific label, for instance, adding a negator for contradictions. That is, when asked to write a contradictory statement against a premise, the annotators often opt to a simple

pattern: negate the premise. Therefore, if the model predicts "contradiction" whenever there is a negator in the hypothesis, it will likely achieve a much-higher-than-chance performance, without actually learning the more complex reasoning skills for NLI. This is indeed the case for several NLI corpora, where the chance-performance is around 33%, but the hypothesis-only baseline is around 60% (Samuel R. Bowman et al., 2020; Poliak et al., 2018).

Another influential study (McCoy et al., 2019) discovered other types of biases, which they named as three heuristics: lexical overlap, sub-sequence overlap, and constituent overlap. The authors constructed NLI examples targeting these three heuristics and named their dataset the HANS corpus (Heuristic Analysis for NLI Systems). Their first heuristic of "lexical overlap" states that if the premise and hypothesis have high lexical overlap (*a dog chases the cat*; *the dog chases a cat*), then this is likely to be an entailment pair. Since most NLI pairs with such a heuristic in SNLI/MNLI are indeed entailment examples, a neural model trained on SNLI/MNLI will adhere to such a heuristic just to achieve higher performance. But this heuristic is clearly problematic, as one can easily think of examples it fails: *a dog chases a cat* CONTRADICTS *a cat chases a dog*. Thus training neural models on such datasets will result in models that take advantage of the heuristics, but incapable of real language understanding. The other two heuristics are "sub-sequence" and "constituent" overlap, which are special cases of the "lexical overlap" heuristic. Sub-sequence/constituent overlap are defined as the cases where the hypothesis is a sub-sequence/constituent of the premise. Based on these definitions, we can see a containment relation: *constituent* $\in$ *sub-sequence* $\in$ *lexical overlap*. Examples from the English HANS corpus are shown in Table 2.8. In chapter 6.3, we follow this work to create a Chinese HANS corpus that targets two of these heuristics in the OCNLI dataset, introduced in chapter 5.

**Attempts to Resolve the Two Issues** To reduce the biases and create datasets with a more difficult evaluation set and/or a better training data, there have been several recent attempts

| Heuristic | Definition | Example |
|---|---|---|
| Lexical overlap | Assume that a premise entails all hypotheses constructed from words in the premise | **The doctor** was **paid** by **the actor**. ŸŸŸŸŸÑ WRONG The doctor paid the actor. |
| Subsequence | Assume that a premise entails all of its contiguous subsequences. | The doctor near **the actor danced**. ŸŸŸŸŸÑ WRONG The actor danced. |
| Constituent | Assume that a premise entails all complete subtrees in its parse tree. | If **the artist slept**, the actor ran. ŸŸŸŸŸÑ WRONG The artist slept. |

Table 2.8: The three heuristics summarized in the English HANS dataset (McCoy et al., 2019), along with examples of incorrect entailment predictions that these heuristics would lead to. Definitions and examples taken from the HANS paper. See Table 6.3 for examples in our Chinese HANS corpus.

in data collection in NLI and also other tasks of NLU, either by improving the instructions given to the annotators (Samuel R. Bowman et al., 2020; Parrish et al., 2021; Vania et al., 2020), filtering out the examples containing biases after data collection (Gururangan et al., 2018; Le Bras et al., 2020; Sakaguchi et al., 2020), or collecting the data in an adversarial setting (Nie et al., 2020a; Potts et al., 2020).

We review two studies for NLI data collection that were conducted concurrently as the work reported in chapter 5. Samuel R. Bowman et al. (2020) experimented with four variants of the vanilla MNLI-style data collection procedure (where an annotator is presented with a premise and asked to write three premises, one for each inference relation), which they call the BASE method: PARAGRAPH, where the premise shown to the annotator is a long paragraph instead of a sentence, EDITPREMISE and EDITOTHER, where the annotator either edits the premise or another sentence that is similar to the premise to make a hypothesis, and CONTRAST, where the annotator writes hypotheses that "show some specified relationship (entailment or contradiction) to a given premise, but do not show that relationship to a second similar distractor premise." (Samuel R. Bowman et al., 2020). Their results show that the conditions they experimented with can greatly reduce the hypothesis-only bias by at least 10 percentage points. However, the results on transfer learning, i.e., training a transformer model on the NLI data first before training it further on

other NLU tasks (including machine-reading task and other NLI tasks) in the SuperGLUE benchmark (A. Wang et al., 2019), do not show any advantage of their proposed methods. That is, all their intervention conditions (PARAGRAPH, EDITPREMISE, EDITOTHER and CONTRAST) yielded worse transfer performance than the BASE method, by about 2 percentage points on the SuperGLUE tasks.

In a similar vein, Vania et al. (2020) experimented with methods for hypotheses collection that do not involve crowdworkers writing the sentences. Specifically, they used two automatic methods to pair up sentences as the premise-hypothesis pair: (1) a similarity based method where sentences from a large corpus with the most similar vector representation are paired up (SIM condition), and (2) machine-translate non-English sentences from an aligned corpus to English and pair the them with their corresponding English sentence (TRANSLATE condition). These pairs are then presented to crowdworkers for inference labelling, i.e., to receive one of the *entailment*, *neutral* and *contradiction* labels. They collected an equal amount of data under the SIM, TRANSLATE and BASE (which is the vanilla MNLI-style data collection) conditions, and compare models trained on them in out-of-domain datasets: MNLI and ANLI, as well as their transfer learning ability on several related tasks such as RTE (Dagan et al., 2005). Just like Samuel R. Bowman et al. (2020), they found no advantage of the models trained on the SIM and TRANSLATE data. In fact, BASE-trained models outperforms them by 2 and 6 points respectively when evaluated on the MNLI and ANLI eval set, and also 3-5 points in the transfer-learning experiments. They did not evaluate on the GLUE diagnostics or other probing datasets, however.

The above work that address the two annotation issues are carefully considered in chapter 5 and chapter 6. Specifically, in chapter 5, we collect the first NLI corpus for Chinese while trying to avoid the two issues mentioned above. However, our priority is to increase the difficulty of the data (fixing issue one), and we monitor the biases mentioned above closely (issue two).

On the other hand, our work has also inspired other work in the field. In a more

recent study, Parrish et al. (2021)—building on the work reported in chapter 5 of this dissertation—carefully compared conditions where linguists actively intervene in the hypotheses-writing process in different forms, either by banning the use of specific words, or using Slack channels to discuss the annotations with the annotators. Their results show that having linguists in the data collection loop to guide the annotators generally reduces the biases and creates more difficult evaluation data. However, using the collected data as training data does not improve neural models' performance on out-of-domain NLI benchmarks.

### 2.1.5 NLI Datasets in Other Languages

While abundant NLI resources exist in English, only very few have been created for other languages, a common situation in NLP research.

**XNLI: The Cross-Lingual NLI Corpus** The first NLI dataset in non-English languages is XNLI (Conneau et al., 2018b), which first collected another 7,500 NLI examples in English using crowd-sourcing, following the MNLI procedure, and then hired translators (from a crowd-source platform) to translate the 7,500 examples into 15 languages. The parallel corpus of 7,500 15 examples then becomes the development and test sets of XNLI. For the training data, the authors used an in-house machine translation (MT) system at Facebook to translate the training set of MNLI into 15 languages. Such a setup allows the authors and other researchers to conduct interesting experiments under several situations: 1) zero-shot or few-shot cross-lingual evaluation (train a model on the English data, and optionally a few training examples in language X, and evaluate on language X), 2) TRANSLATE-TRAIN (train with the machine-translated training data in language X, and test on the evaluation data in X, which is human translated), 3) TRANSLATE-TEST (translated the examples during test time to the language the model is trained on, for instance English, using an MT system). At the time of the release of XNLI, the second and

| | Premise | Hypothesis |
|---|---|---|
| 1 | Louisa May Alcott Œ Nathaniel Hawthorne O ( Pinckney W S £ 个 « Oliver Wendell Holmes ð 为“ t ) W S „ Beacon Street W S O @ 些 œ " ê 9 ê 擂 „ † ò f ¶ William Prescott<br>*Eng.: Louisa May Alcott and Nathaniel Hawthorne lived on Pinckney street, but on Beacon Street street, which is named “Sunny Street by Oliver Wendell Holmes, lived the bragging historian William Prescott. [sic]* | Hawthorne O ( Main Street上<br>*Eng.: Hawthorne lived on Main Street.* |
| 2 | 东 ¹ „ Passeig de Gracia y + / Diputacie Consell de Cent Mallorca Œ Valancia ô 0 Mercat de la Concepcie :<br>*Eng.: Look at the Passeig de Gracia from the East, especially Diputacie, Consell de Cent, Mallorca and Valancia, up to Mercat de la Concepcie market.* | : ú . ' Ï „ 4 œ Œ , Ü<br>*Eng.: The market sells a lot of fruit and vegetables.* |
| 3 | Ð L Slient Ð L Deep Ð L T H<br>*Eng.: run Slient, run Deep, run answer. [sic]* | „ „ „ p<br>*Eng.: secretly escape.* |
| 4 | 下一 6 µ 中 Å @ ú 0 Ä Ç „ # 人 S ö Þ Æ ô 他不 ¤ 为他 „ L # / ü ” å Z什么 不 ” å Z什么<br>*Eng.: In the next phase, the CIA director on Al Qaeda recollected that ...* | ü ¤ 为 Ù Œ h Ö ³ 于他<br>*Eng.: The (movie) director thought ...* |
| 5 | à d s G 上 Webster „ ° ' f Í x Ø 8 ( Í x Ô Webster „ ° 世 L Í x Œ Ž ý W 产 Í x ó ~ 之50 „ Í a Œ á o Ï<br>*Eng: Thus, on average, the Webster New Collegiate Dictionary ...* | f ! • ( æ / ¯ y f b à 为 ƒ / } „<br>*Eng: The school uses the Webster College, because it is the best* |

Table 2.9: Examples sampled from 200 NLI pairs we manually checked in the crowd-translated XNLI development set (in Chinese) (Conneau et al., 2018b). Translations are provided by us. 1 and 2: problems of *translationese*, too many untranslated proper names. 3: incomprehensible example. 4 and 5: poor translation quality. In 4, the CIA director is translated as “movie director” ( ü ) in the hypothesis. In 5, the Webster New Collegiate Dictionary is translated as “Webster college” ( æ / ¯ y f b ) in the hypothesis.

third methods have better performance. However, more recently, there has been enormous progress in cross-lingual transfer of multilingual models (Conneau et al., 2020), which will be explained in chapter 2.2.2.

While automatically translated data have proven to be useful in many contexts, such as cross-lingual representation learning (Siddhant et al., 2020), there are well-known issues, especially when used in place of human annotated, quality controlled data. One issue concerns limitations in the quality of automatic translations, resulting in incorrect or unintelligible sentences (see examples from the crowd-translated XNLI dataset in Table 2.9). Specifically, in example 4 in Table 2.9, the CIA director is translated as “movie director”

( ü ) in the hypothesis. In in example 5 in Table 2.9, the Webster dictionary is translated as "Webster college" ( æ / ¯ y f b ) in the hypothesis.

But even if the translations are correct, they suffer from "translationese", resulting in unnatural language, since lexical and syntactic choices are copied from the source language even though they are untypical for the target language (H. Hu and Kübler, 2020; H. Hu et al., 2018; Koppel and Ordan, 2011). A related issue is that a translation approach also copies the cultural context of the source language, such as an overemphasis on Western themes or cultural situations, exemplified in the first two examples in Table 2.9, where many English names are directly carried over into the Chinese translation. Other aspects of English syntax, such as long relative clauses, which are common in English but dispreferred in Chinese (C.-J. C. Lin, 2011) may also be carried over in a translated dataset.

**Monolingual NLI datasets in Chinese and other languages** In fact, there has been relatively little work on developing large-scale human-annotated resources for languages other than English, as reviewed in chapter 2.1.

For Chinese, the only available corpora are XNLI (Conneau et al., 2018b), which consists of training data machine-translated from the English MNLI and dev sets that are crowd-translated from English, and CMNLI, which is also machine-translated (MT) from MNLI but using a different MT system, published in the first version of the CLUE benchmark (Xu et al., 2020).[9] To the best of our knowledge, the resource reported in chapter 5 and chapter 6 are the only non-translated Chinese NLI resources created so far.[10]

For other languages, there exist only a few NLI datasets, for instance Fonseca et al. (2016) and Real et al. (2020) for Portuguese, Hayashibe (2020) for Japanese, Wijnholds and Moortgat (2021) for Dutch, and Amirkhani et al. (2020) for Persian, but none of them have human elicited sentence pairs. Hayashibe (2020), Fonseca et al. (2016) and Amirkhani et al. (2020) used automatic methods to pair sentences into hypothesis-premise pairs and then

[9]The CMNLI corpus can be downloaded from https://github.com/CLUEbenchmark/CLUE.

[10]There are other unpublished machine-translated Chinese NLI resources on Github such as https://github.com/blcunlp/CNLI, which is also machine-translated from subsets of SNLI and MNLI.

asked human annotators to label them. Other efforts have focused on automatic translation of existing English resources (Mehdad et al., 2011), sometimes coupled with smaller-scale hand annotation by native speakers (Agić and Schluter, 2017; Negri et al., 2011). For instance, Real et al. (2020) and Wijnholds and Moortgat (2021) are both based on (machine-)translated versions of the English SICK dataset. This is also true for some of the datasets included in the first version Chinese NLU benchmark CLUE (Xu et al., 2020) and for XNLI (Conneau et al., 2018b), a multilingual NLI dataset covering 15 languages including Chinese.

We summarize the current non-English NLI datasets in Table 2.10. We can see that many of the resources are created using either machine or human translation, rather than built from scratch.

### 2.1.6 Cross-lingual Benchmarks in NLU

As we will introduce in chapter 2.2.2, a recent trend of neural modeling is multilingual models where a single model is trained to perform NLP tasks in many languages. In order to measure the effectiveness of these models, we need multilingual or cross-lingual benchmarks that can simultaneously evaluate a model on many languages, for example the XNLI corpus, which has a parallel dev set for 15 languages where all the NLI pairs are translated from the English pairs. The parallel setting in XNLI allows the results on different languages to be comparable.

Apart from XNLI, there has also been many other efforts building multilingual datasets and benchmarks. For example, PAWS-X for cross-lingual paraphrase identification (Y. Yang et al., 2019), EXAMS for cross-lingual question answering of high school exams (Hardalov et al., 2020), MKQA for open domain question answering (Longpre et al., 2020), XCOPA for multilingual causal commonsense reasoning (Ponti et al., 2020), XQuAD for question answering (Artetxe et al., 2020), and XTREME which gathers multiple datasets into a benchmark (J. Hu et al., 2020).

Most if not all the benchmarks mentioned above are created for specific NLP tasks such as question answering. There are very few resources for cross-lingual *probing*. That is, they focus on the transfer ability of a model on a task, rather than probing whether transfer learning is possible for some targeted reasoning skills, or a linguistic/logical phenomenon. We believe that it is also important to know whether the transfer ability is connected with the specific capabilities needed for a task, and thus address the issue of cross-lingual *probing* in chapter 6.[11]

## 2.2 Previous Approaches

To enable *computers* to make inferences, researchers have proposed various ideas and approaches, which we turn to now. For the purpose of this dissertation, we will focus on introducing previous literature on logic-based, symbolic models, and the recent neural models, in particular the pre-trained transformer models.

### 2.2.1 Symbolic Approaches

Entailment has been studied in logic and semantics for a long time. Thus it is natural to build symbolic systems based on research and tools in the logic tradition.

Here we categorize systems into logic-based and natural-logic-based, depending on whether the systems represent natural language input in logical forms (for instance, first-order logic or high-order logic) or in natural logic (which closely mimics the surface forms in natural language).

**Logic-based Systems** Systems in the logic-based approach generally translate input natural language into logical forms and then call theorem provers to find proofs (Bjerva et al., 2014; Kalouli et al., 2020; Martínez-Gómez et al., 2017; Yanaka et al., 2018). For example, Bjerva et al., 2014 used a system that first produces a "formal semantic representation" of

[11] See a review of probing studies in chapter 2.2.2.

sentences, then translates the representation to first-order logic, and finally calls "off-the shelf theorem provers and model builders" to determine the relation between a premise and a hypothesis.

Yanaka et al., 2018 adopts Neo-Davidsonian event semantics as their meaning representation, which is then represented as directed acyclic graphs (DAGs). They use a theorem prover to perform unification of subgraphs, and then the proof is based on basic formulas and inferences rules of natural deduction. Because of the graph representation of sentences and the unification operation, their system can handle some syntactic variation. At the moment, their system achieves the state-of-the-art results on SICK (M. Marelli et al., 2014) for logic-based models (84.3% in accuracy).

Logic-based models commonly have high precision for entailment and contradiction, because to return an entailment or contradiction prediction, the models need to find a formal proof in the system. These models can thus make good use of the theorem provers that have been studied and researched for a long time. However, one major bottleneck is the translation of natural language into logical forms. For example, while it is easy to translate *every dog walks* to $@x(dog(x) \rightarrow walk(x))$, many sentences in natural language are very difficult to be translated into logical forms. MacCartney and Christopher D Manning (2008) gives an example where there is probably no satisfactory translation of *Every firm saw costs grow more than expected, even after adjusting for inflation* into first-order logic.

Natural logic, on the other hand, has a syntax that resembles the syntax of natural language, and thus does not need the error-prone translation from sentences to logical forms. We turn to natural logic now.

**Natural Logic** Rooted in Aristotelian syllogism, natural logic is a logic system that mimics the surface forms of human language (van Benthem, 2008). Thus there is no need to translate sentences into any logical forms.

For instance, in the natural logic fragment on the quantifier *all* (A) and relative clauses

(RC), ApRCq, we have the following axioms/rules (Moss, 2018).

$$\frac{}{\text{All } x\ x}\ \text{AXIOM} \qquad \frac{\text{All } x\ y \qquad \text{All } y\ z}{\text{All } x\ z}\ \text{BARBARA} \qquad \frac{\text{All } x\ (r\ \text{all}\ y) \qquad \text{All } z\ y}{\text{All } x\ (r\ \text{all}\ z)}\ \text{DOWN}$$

In the above logical system, $x$ and $y$ are nouns, and $r$ are relations (or transitive verbs). Therefore, the syntax "All $x$ $y$" in this logic has the same meaning as it has in natural language. For instance, "All *beagles* *dogs*" simply means: all beagles are dogs. The same is true for "All $x$ ($r$ all $y$)", an example of which could be "All *dogs* (*love* all *cats*), meaning: all dogs love all cats.[12] The AXIOM then states the obvious fact that all entities are a subset of themselves (all dogs are dogs), while BARBARA is the classic Aristotelian syllogism, for instance, if all beagles are dogs and all dogs are animals, then we know that all beagles are animals. Finally DOWN states that if all $x$ has a relation $r$ with all $y$, and $z$ is a subset of $y$, then we know that all $x$ has the same relation $r$ with all $z$. This is also easy to see, because $z$ is a subset of $y$.

Of course, these three rules should be proved based on the semantics of the "All $x$ $y$" and "$r$ all $x$", defined in model-theoretic terms. However, the beauty of natural logic is that its syntax is so close to the syntax of natural language that one can perform the reasoning using natural language alone, once a set of rules like the above have been obtained and proved.

Central to the natural logic tradition is the phenomenon of **monotonicity**. Concretely, each quantifier has a monotonicity profile. For example, "all" is antitone (or downward entailing, denoted by $^{\downarrow}$) in its first argument and monotone (or upward entailing, denoted by $^{\uparrow}$) in its second argument. That means, for "all dogs$^{\downarrow}$ are walking$^{\uparrow}$", we can replace the first argument with words of a smaller extension: "all dogs are walking" entails "all *beagles* are walking". Conversely, we can replace the second argument with words of a larger extension: "all dogs are walking" entails "all dogs are *moving*". In fact, AXIOM, BARBARA

[12] To simply things, we define $x$ and $y$ as nouns, but they can be defined recursively as terms involving relations, as in "$r$ all $y$", e.g., *see* all *dogs*, denoting all the entities that see all dogs.

and DOWN can be uniformly interpreted under monotonicity and explained/implemented by a single replacement operation based on monotonicity. For BARBARA, from the second premise "All $y$ $z$", we know that $z$ has a larger extension (or formally $y$ ⌑ $z$). Then from the first premise "All $x^{\acute{0}}$ $y^{\grave{0}}$", which is now augmented with monotonicity information, we can straightforwardly replace $y$ with $z$, and obtain the entailed statement: "All $x$ $z$". We can obtain the result in DOWN in a similar manner. Note that the monotonicity property when two "all" in the first premise in DOWN should be: "All $x^{\acute{0}}$ ($r^{\grave{0}}$ all $y^{\acute{0}}$)". Thus we can replace $y$ with $z$ which has a smaller extension. See chapter 3.2.2 and chapter 4.2 for more details on monotonicity.

Natural logic and monotoncity reasoning have received much attention in recent years in the field of NLI/RTE (MacCartney, 2009; MacCartney and Christopher D Manning, 2008, 2007; Yanaka et al., 2019a,b) as well as logic and linguistics (Deng et al., 2020; Icard, 2012; Icard and Moss, 2013, 2014).

For symbolic systems in NLI, the appeal of natural logic is that it bypasses the translation from natural language into logical forms, and that the resemblance between natural logic axioms/rules and natural language will allow the system to rely on surface forms. Finally, the natural logic proof will also be easier to interpret, as it does not involve complicated logical forms.

Now, we review two systems (`NatLog` and `LangPro`) that are most relevant to the symbolic system proposed in this dissertation. Both rely on natural logic and monotonicity but in different ways.

**The `NatLog` System** The `NatLog` system in MacCartney, 2009; MacCartney and Christopher D Manning, 2008, 2007, 2009 starts with a *polarized* premise, i.e., a premise tagged with monotonicity arrows ($\acute{0}, \grave{0}$). To determine the relation between a premise and a hypothesis, `NatLog` 1) finds the atomic edits from the premise to the hypothesis, 2) computes the polarity of each edit and 3) uses a "join" operator that returns the final relation aggregated

from all the edits.

Specifically, NatLog defines three atomic edits INS (insertion), DEL (deletion) and SUB (substitution). It also defines seven entailment relations between sets (MacCartney, 2009, pp. 79), including equivalence (couch ≡ sofa), forward entailment (dog ⊏ animal), backward entailment (furniture ⊐ chair), negation (kind ^ unkind), alternation (cat | dog), cover (animal ⌣ non-ape)[13] and independence (hungry # chair).[14] Each atomic edit will result in a lexical relation and then based on the context a projected relation can be obtained. Combining the relations according the sequence of atomic edits via a "join" operation will produce the final entailment relation between the premise and the hypothesis.

For example, if we have a $P$: *every dog*[0] *dances* and a $H$: *every poodle dances*, to go from $P$ to $H$ involves only one edit, SUB(*dog, poodle*), i.e., substituting *dog* with *poodle*. Then since the lexical relation between *dog* and *poodle* is backward entailment (⊐), and crucially *every* assigns a downward monotonicity to its first argument *dog*, the lexical relation is flipped to forward entailment (⊏). This is the correct result since $P$ indeed entails $H$ in this example.

Because of the computation of atomic edits which basically *aligns* the premise and the hypothesis, NatLog is able to handle syntactic variation to a certain extent. Empirically, NatLog is the first system to be tested on the FraCaS (Cooper et al., 1996) dataset and has been shown to have very high precision. It also improves accuracy on the RTE dataset (Dagan et al., 2005) when hybridized with a machine-learning inference engine (MacCartney and Christopher D Manning, 2008).

While NatLog is the first system to perform the polarization task, there are several problems with it. One is that since their polarization algorithm is based on heuristics, it can lead to errors if the heuristics do not cover a monotonicity phenomenon. We tested their system as part of Stanford CoreNLP v3.9.1. For downward entailment operators such as *refuse* and *without*, the system seems to incorrectly polarize all the words as upward entail-

[13] In set-theoretic terms, $\wp x \subset y \quad \forall q \supset \wp x \supset y \quad Uq$.
[14] Also see chapter 1.1.1.

ment: *Ed*$^{\downarrow}$ *refused*$^{\downarrow}$ *to*$^{\downarrow}$ *dance*$^{\downarrow}$; *John*$^{\downarrow}$ *loves*$^{\downarrow}$ *dancing*$^{\downarrow}$ *without*$^{\downarrow}$ *shoes*$^{\downarrow}$ (*walks* and *dance* should receive $^{\downarrow}$). Another issue is that `NatLog` is unable to produce the polarity for words that are neither upward or downward entailing, e.g., *I*$^{\downarrow}$ *like*$^{\downarrow}$ *most*$^{\downarrow}$ *dogs* . A more systematic evaluation of `NatLog` and our proposed system `ccg2mono` will be conducted in chapter 3 of the dissertation. Another problem with `NatLog` is that different edit sequences might lead to different results (see section 6.5.5 of MacCartney, 2009). This is potentially a more serious issue, as `NatLog` might give different predictions for the same premise-hypothesis pair[15].

`NatLog` operates on surface forms of the premise and thus differs from the above mentioned logic-based approaches (Bjerva et al., 2014; Yanaka et al., 2018).

**The `LangPro` System** The `LangPro` system (Abzianidze, 2014, 2015, 2016a, 2017) on the other hand, does translate the input text into $\lambda$ terms commonly used in formal semantics and then operates on them.

`LangPro` is composed of several parts: a Combinatory Categorial Grammar (CCG) parser, a LLFgen (Lambda Logical Forms generator), a LLF-aligner, a prover and a Knowledge Base. The CCG parser first produces a CCG parse tree, which is then translated to $\lambda$ logical forms by the LLFgen. A LLF-aligner is optionally used to align identical chunks of LLFs in the premise and hypothesis and thus these chunks will be considered as a whole without any internal structures. Finally, a theorem prover based on a first-order logic prover (Fitting, 1990) makes the final decision. The rule inventory of the prover contains "about 50 rules", which are manually coded (see Abzianidze, 2014, for details of the rules). [16]

`LangPro` has achieved state-of-the-art precision (97.95%) and competitive accuracy (81.35%) on the SICK dataset (Abzianidze, 2015).

While `NatLog` and `LangPro` have been successfully applied to NLI/RTE datasets,

[15] Although in section 6.5.5 of MacCartney, 2009, he reported to have found no *unsound* predictions, only undetermined predictions. That is, in one sequence, `NatLog` returns the correct relation; in another, it returns unknown.

[16] See Abzianidze, 2015 Sections 2, 3, 4 for details.

one issue is that they are rather complex. Can we build an inference system based on natural logic, but is much more light-weight? If the answer is yes, how far can this system go? That is, how much of natural logic related inference can it cover, given that it is light-weight and more straightforward than the above two systems? These question will be addressed in chapter 3 and chapter 4.

### 2.2.2 Neural Approaches

In this section, we present a brief overview of the neural models used in NLI. However, as the field is so large that many dissertations and books have been written about it, we will not be able to cover everything. The focus will be on the pre-trained transformer encoders such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). For a more detailed review, interested readers can refer to Goldberg (2016) and Rogers et al. (2020).

**Word Embeddings** The neural approach first became popular in NLP with the introduction of word embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). The idea is to represent words as vectors, rather than strings in a symbolic system. This idea goes back to the Latent Semantic Analysis in information retrieval where each document is represented by a low-dimension vector and document similarity is computed from the cosine similarity of their corresponding vectors (Deerwester et al., 1990). To obtain a latent, vector representation of the words, different methods have been proposed: learning from PMI matrices, where each cell of the matrix denotes the point-wise mutual information between any two words in the vocabulary (Church and Hanks, 1990; Turney and Pantel, 2010), the Skip-gram with negative sample (SGNS) variant of the word2vec model, where the weights of a classifier learning to predict whether $word_i$ is in the context of $word_j$ are used as word representations (Mikolov et al., 2013), the Global Vector (GloVe) which is based on matrix factorization techniques performed on word-context co-occurreence matrix obtained from a large corpus (Pennington et al., 2014), among oth-

ers. While the particular algorithms used for these algorithms are different, the theoretical foundation for them is the same hypothesis in distributional semantics: you shall know the word by the company it keeps (Firth, 1957). That is, based on co-occurrence information of words in a raw corpus, we can already extract useful information about the semantics of the words.

The above ideas on word embeddings are all first proposed on English. For Chinese, the idea of word embeddings has been widely used in Chinese NLP applications as well. Several word embedding models have been proposed, by treating different linguistic units as the tokens to learn embeddings for, for instance, Chinese characters, $n$-grams, or even phrases (S. Li et al., 2018; Song et al., 2018).

**Sentence Embeddings** While word embeddings are about the representation of word meaning, neural network architectures such as recurrent neural network (RNN) are commonly used to obtain the compositional meaning of a sentence from the tokens that form the sentence. In other words, we are interested in obtaining a good vector representation of a sentence/text, based on the vectors of the words in that sentence/text. Common ways obtaining the sentence representation include: averaging, summing, pooling, using tree structures, etc., or some combinations thereof. For instance, averaging means that we first perform element-wise addition of the vectors of the $n$ words in the sentence, and then divide the resulting vector by $n$, which will give us the sentential representation. These methods have served as the baselines for more complicated sentence embeddings.[17]

In chapter 5.4, we experiment with several such neural architectures on our collected Chinese NLI corpus, following the baselines experimented for MNLI.[18] The first model is a continuous bag-of-words model (**CBOW**) where the premise and the hypothesis are both represented as the sum of the embeddings of their individual words, which we denote as

[17] See the baselines of MNLI at `https://github.com/nyu-mll/multiNLI`.
[18] `https://github.com/nyu-mll/multiNLI`

$V_{prem}$ and $V_{hypo}$. The sentence pairs are then represented as

$$\mathsf{r}V_{prem}; V_{hypo}; V_{prem} \quad V_{hypo}; V_{prem} \quad V_{hypo}\mathsf{s}$$

where the colon denotes vector concatenation, $V_{prem} \quad V_{hypo}$ denotes element-wise difference and $V_{prem} \quad V_{hypo}$ denotes element-wise multiplication. The latter two operations are assumed to capture the similarity of the premise and the hypothesis, and have been shown to be empirically effective in previous modeling work on natural language inference (Mou et al., 2016). This is passed on to a 3-layer neural network. The second model is a **biLSTM**, where the mean of the hidden states of a bidirectional LSTM is used for sentence representation. The sentence-pair is also represented as the concatenation of 4 vectors above, which is then passed on to an MLP for classification.

Next we introduce the BERT model, which has been shown to outperform the neural models using static word embeddings and various forms of RNNs.

**BERT** Bidirectional Encoder Representations from Transformers (BERT) is a neural model introduced in Devlin et al. (2019).[19] When introduced, it produced new state-of-the-art results on many NLI tasks. Since then, it has inspired a large body of work improving on the model architecture, applying it to different downstream tasks, and exploring the inner workings of the model itself, forming almost a new field of study that some referred to as BERTology (Rogers et al., 2020). The BERT model builds on several ideas in the neural tradition of NLP: contextualized embeddings (McCann et al., 2017; Peters et al., 2018), transformer architecture (Vaswani et al., 2017), bidirectional language models, sub-word tokenization, among others. In the paragraphs below, we will introduce the main ideas behind BERT. BERT and its variants are the main neural models we experiment with in chapter 5 and chapter 6.

At the highest level, BERT is a neural model pre-trained with a masked language mod-

[19] https://github.com/google-research/bert

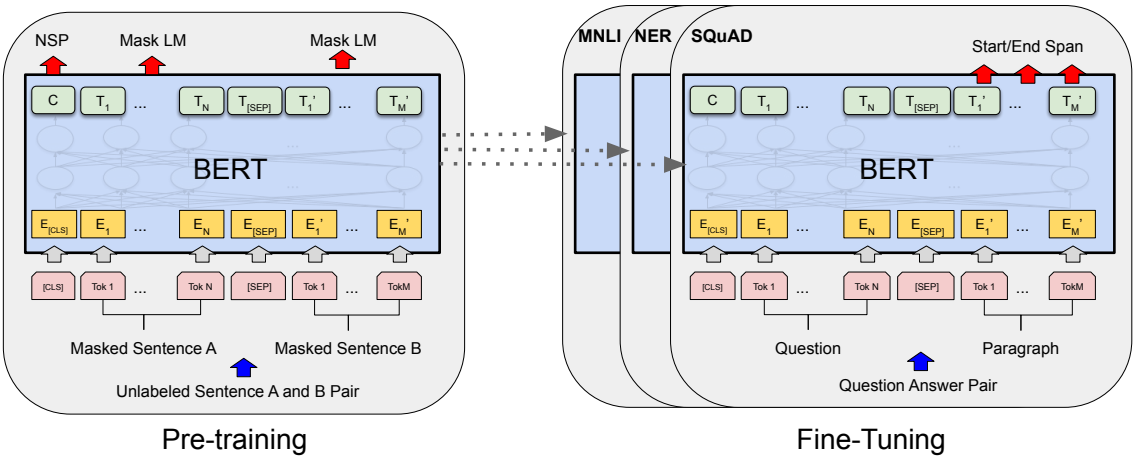



Figure 2.1: Two training steps of BERT: pre-training on raw text corpora and fine-tuning on target tasks, image taken from Devlin et al. (2019).

eling objective (predict the words in the blanks of sentences) and a next sentence prediction object (whether $\text{sentence}_1$ precedes $\text{sentence}_2$), and is then fine-tuned on downstream tasks (see Figure 2.1 for the two training steps). BERT can perform sentence(-pair) classification task, as well as sequence labeling task. Next, we go into details of the critical components of BERT.

The first important feature of BERT is "contextual embeddings" where a word will be encoded with different embeddings in different context, an idea pioneered in the CoVe (Contextualized Word Vectors) (McCann et al., 2017) and ELMo (Embeddings from Language Models) embeddings (Peters et al., 2018). As a reminder, the classic word2vec and GloVe embeddings are known as static word embeddings in that each token in the vocabulary has only one representation. The obvious issue is that many words in natural language are polysemous, and it would not be ideal to use the same vector to represent the different meanings of the same word. For instance, "bank" as a financial institution and "bank" as the river bank should ideally have two different representations (which should also be far away in the embedding space since the two meanings have little relation to each other). In BERT, the embedding of a word depends on the context it is in. That is, every instance of a word will have a different embedding that is determined in relation to the embeddings of

other words in the context. The idea of contextualized embeddings has played a crucial role in all the subsequent variants of BERT, such as RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), etc.

The second important component of BERT is the self-attention mechanism. BERT uses the encoder from the Transformer architecture (Vaswani et al., 2017). The idea is that there are weights in the neural network about how much attention each token should pay to all the tokens in the same input text. This is intended to allow the model to better handle long distance dependency (than an RNN architecture), since in a transformer model, there are designated weights for every (*token*, *token*) pair of the input text, forcing the model to explicitly learn the dependency between even the first token and the last token of the input.

The third component of BERT is the pre-training task. Unlike word2vec models which are trained with the objective of predicting whether $\text{word}_1$ is in the context $\text{word}_2$, or a regular language model whose objective is to predict the next word, BERT's training objectives are: 1) masked language modeling, also called Cloze-test (Taylor, 1953) and 2) next sentence prediction. Both are self-supervised training methods in that they can be performed on large quantities of raw text without needing any human annotation. In the masked language modeling, the model's objective is to predict some randomly masked words (i.e., blanks) in the sentence, based on the context: *John was running very ___ since he doesn't want to miss the bus*. This task allows the model to attend to both the left and right context. In next sentence prediction, the model is presented with two sentences and asked to predict whether the second one is a continuation of the first one. This allows the model to learn the relation between sentence pairs, which is assumed to help sentence-pair classification tasks such as NLI.[20] Now that the model has been pre-trained, it will then be further trained, or fine-tuned, on the target task. The fine-tuning step will follow the common supervised training procedure in machine learning. In all our experiments, we are fine-tuning BERT

[20]In other variants of BERT such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019), the masked language modeling training is kept, but the next sentence prediction is either removed or replaced with other training objectives.

on the NLI datasets, rather than pre-training it. The first figure in Figure 2.1 depicts the pre-training stage of BERT.

In terms of the pre-training data, BERT is pre-trained on the BooksCorpus (Zhu et al., 2015) and Wikipedia, which has a total of 3,300 million words (Devlin et al., 2019). Thus, at a high-level, BERT can be said to have "read" a gigantic amount of raw text (with blanks), and is trained to 1) fill the blanks and 2) predict whether a randomly sampled pair of text from the corpus is originally next to each other in the corpus. Then for each downstream task, for instance NLI, BERT will be further fine-tuned in a supervised manner, as indicated in the second figure in Figure 2.1.

With respect to performance, at the time of release, BERT achieves new state-of-the-art results on 11 tasks, many with large margin over the previous best results. It has inspired a long line of work on pre-training transformer models, with notable examples such as the RoBERTa (Robustly Optimized BERT) (Liu et al., 2019), ALBERT (A Lite BERT) (Lan et al., 2019), DistilBERT (distilled version of BERT) (Sanh et al., 2019), BioBERT (biological language BERT) (J. Lee et al., 2020), XLNet (Z. Yang et al., 2019), among many others.

Given the success of the BERT-family models, in chapter 5 of this dissertation, BERT and its variants are the major neural architectures we experiment with. Devlin et al. (2019) also released a Chinese BERT and a multilingual BERT, along with the English BERT.[21] The experiments in chapter 5 and 6 are run with the Chinese BERT and the enhanced Chinese RoBERTa published in Cui et al. (2019).

**XLM-RoBERTa** One claimed advantage of the neural approaches in NLP is the universality of vector representation across languages. That is, if text in every language is represented as vectors, then we can perform any computation in the "universal" language of vectors, without the need to worry about the differences among different human languages, which is a major obstacle in symbolic approaches. Specifically, if we can obtain some multilingual word embeddings such that the vectors representing the English word

[21] https://github.com/google-research/bert

*cat* and the Chinese word 猫 are close in the semantic space, then we have found an abstract representation of the concept of a cat, independent of the specific language in which the concept is encoded in.

To achieve such a goal, a common strategy is to expand the pre-training data (significantly) to include many more languages. At the same time, one also needs to expand the vocabulary of the model to include the tokens of all the languages in the pre-training data. For instance, the recent XLM-RoBERTa model, which is based on RoBERTa, is pre-trained with 2.5TB of text from 100 languages including Chinese, using the masked language modeling objective (Conneau et al., 2020). The pre-training data are from a CommonCrawl Corpus (crawled from the web) prepared by the authors of the same paper. In order to handle the orthography of all 100 languages, XLM-RoBERTa (XLM-R henceforth) uses a large vocabulary of size 250k, compared to the typical vocabulary size of 20-30k for a monolingual model (Devlin et al., 2019).[22]

Once the model is pre-trained on 100 languages, in the second fine-tuning step, it only needs to be fine-tuned on human labeled data in one language, and then can (theoretically) be tested on test sets in all the 100 languages it is pre-trained on. To give an example, which will be discussed in chapter 6, if we fine-tune XLM-R on English NLI data alone (such as MNLI), we should expect it to perform reasonably well on Chinese NLI, even though it has not seen a single NLI pair in Chinese! This is referred to as zero-shot, cross-lingual transfer learning. XLM-R is the first model to have competitive performance in both scenarios above. In particular, it gains more than 10% increase in performance on cross-lingual tasks such as XNLI (Conneau et al., 2018b) and MLQP (P. Lewis et al., 2020) over the multilingual BERT, and for the first achieves such high performance without

[22]Note that, in all the transformer models, specifically-designed algorithms are used to perform vocabulary construction and tokenization, e.g., byte-pair encoding (Sennrich et al., 2016), WordPiece (Devlin et al., 2019; Y. Wu et al., 2016), and unigram language model tokenization (Kudo, 2018). For these algorithms, the smallest linguistic unit is not a word, but usually some "sub-word" unit (which may resemble a morpheme), determined by the algorithm. For instance, in the WordPiece tokenizer for BERT, the word "embeddings" is tokenized into four sub-words: em, ##bed, ##ding, ##s, where "##" indicates that it is part of the previous token.

sacrificing its performance on English NLU benchmarks, having competitive results to strong monolingual English models on the GLUE benchmark (Conneau et al., 2020).

Several other studies have also shown the surprisingly good performance of XLM-R in zero-shot transfer conditions, for a number of languages, Khashabi et al. (2020) for Persian, Choi et al. (2021) for Korean, S. Wu and Dredze (2019) for 39 languages on 5 NLP tasks, among others. In addition to studies cited above, positive results of cross-lingual transfer across a wide range of languages are reported in Nozza et al. (2020) and S. Wu and Dredze (2020), with a focus on transfer across specific tasks such as POS tagging, NER.

There is also a series of work trying to understand how and why cross-lingual transfer works. For example, S. Wu and Dredze (2019) and Pires et al. (2019) probes the multilingual BERT model (mBERT, Devlin et al. (2019)) by either examining the representation of different layers in the transformer architecture, or the lexical overlap of different languages in mBERT's vocabulary. S. Wu and Dredze (2019) show that: all layers of mBERT preserves language-specific information, in that the weights from the layers can be used to perform language identification with high accuracy. They also found "a strong correlation is observed between the percentage of overlapping subwords and transfer performance", namely better performance when there is larger vocabulary overlap. Pires et al. (2019) examines the cross-lingual transfer learning ability of mBERT for language pairs based on typological features of the languages. Their most surprising result is that an mBERT model can perform zero-shot transfer between pairs of languages that have no lexical overlap at all: an mBERT fine-tuned using only POS labeled Urdu (written in Arabic script), achieves 91% accuracy on Hindi (written in Devanagari script) under zero-shot condition, evaluated on data from (Zeman et al., 2018). They argue that "this provides clear evidence of mBERT's multilingual representation ability, mapping structures onto new vocabularies based on a shared representation". While mBERT is learning a shared representation, it performs best on typologically similar languages. For instance, transferring from English to Japanese does is less effective than transferring from English to Bulgarian, potentially

due to the fact that English and Bulgarian are both SVO languages while Japanese is SOV. Karthikeyan et al. (2019) investigates model properties such as the network depth and number of attention heads, as well as the effect of language similarity (in terms of word-order and structure) on the cross-lingual transfer performance. Specifically, instead of multilingual BERT, Karthikeyan et al. (2019) pre-trained Bilingual-BERT (B-BERT) for three language pairs, and experimented with cross-lingual transfer learning of B-BERT on NLI and NER. Their results show that vocabulary overlap plays a "negligible" role in cross-lingual success, while structural similarity (such as word order) and model depth play significant role in cross-lingual transfer learning.

While these studies have shed light on some linguistic properties of multilingual models, to the best of our knowledge, no studies have investigated whether cross-lingual transfer can be successful on specific linguistic phenomena which are unique in the target language. Chapter 6 of this dissertation studies this issue by testing the transfer learning performance of XLM-R in the task of Chinese NLI. Specifically, we design various probing datasets to explore the linguistic ability of XLM-R for different linguistic phenomena, following the long line of work on *probing* the neural models, which the above studies on mBERT belong to. we will review the probing studies in greater detail next.

***Probing* studies for the neural models** Given the success of the neural models and the the relative difficulty in interpreting what the models are learning (compared with symbolic models), much effort has been devoted to *probe* the neural models. The term *probe* in this context generally refers to understanding what the neural models are doing, for instance what kind of knowledge is learned after the (pre-)training phase, where and how the different aspects of human language are represented in the neural network, to name just a few.

Therefore, in the broadest sense, any study that is aimed at investigating the representations in neural models can be said to be a *probing* study. However, in the narrow sense,

there are mainly two types of probing studies in the literature (Tenney et al., 2019a). The first is to perform **behavioral studies**—reminiscent to what has been done in psychology and psycholinguistics to the human subject except that the subject of study is now an artificial neural network—where probing datasets are designed in a controlled manner, targeting the linguistic/logical/world knowledge that one plans to probe, for instance, subject-verb agreement (Marvin and Linzen, 2018), monotonicity (Richardson et al., 2020; Yanaka et al., 2019a), function words (Kim et al., 2019), etc. Then the researchers analyze the errors of the models to reverse-engineer what the model has or has not learned. The probing studies mentioned in chapter 2.1.4 and 2.1.1 mostly fall into this category, where the task of NLI is used as the task to probe the representations of the neural models. However, it should be noted that NLI is not the only task adopted/appropriate for probing. In earlier probing studies, other tasks have been designed as well, some with the goal of probing linguistic abilities outside semantics and language meaning, e.g., syntactic information. For instance, Conneau et al. (2018a) constructed "10 probing tasks designed to capture simple linguistic features of sentences" including number of words, constituent tree depth, tense, subject number, etc. to test different models for sentence embeddings. Marvin and Linzen (2018) designed minimal pairs for several syntactic phenomena (subject-verb agreement, reflexive anaphora and negative polarity items) and use the probabilities assigned by an LSTM language model to the sentences in the minimal pair to study whether LSTM models are capturing syntax.

The second method is to directly analyze the representations in the neural architecture "to assess whether there exist localizable regions associated with distinct types of linguistic decisions." (Tenney et al., 2019a). One technique for doing so is the **probing classifiers** (Tenney et al., 2019a,b). Specifically, the neural representations of from a model are first extracted, and are then fed to a probing classifiers as features to perform some of the core NLP tasks, such as part-of-speech tagging, constituent labeling, dependency labeling, semantic role labeling, coreference, etc. The performance of these classifiers on task $t$ is

interpreted as the the amount of information in an embedding for $t$. Tenney et al. (2019b) then compared the contextualized embeddings and their non-contextualized counterparts using the probing classifiers and concluded that the improvement of contextualized embeddings is largely on syntactic tasks. Tenney et al. (2019a) performed a similar study comparing the embeddings from different layers of the BERT model and concluded that the lower layers are associated more with lower-level processing such as part-of-speech tagging and parsing, while the higher layers are associated with semantic processing such as semantic role labeling and coreference.

For XLM-R specifically, there has been work that fall into the second category, which has focused on either examining the representation of different layers in the transformer architecture or the lexical overlap between languages (Pires et al., 2019; S. Wu and Dredze, 2019). Karthikeyan et al. (2019) investigate the role of network depth and number of attention heads, as well as syntactic/word-order similarity on the cross-lingual transfer performance.

Our work in chapter 6 falls into the first category, where we will examine closely the zero-shot transfer ability of XLM-R in Chinese, using four types of probing and adversarial NLI datasets that we constructed. In particular, we are interested in whether XLM-R can successfully transfer on NLI problems involving uniquely Chinese linguistic phenomena, for instance, *pro*-drop, four-character idioms (*chengyu*), etc.

## 2.3 Summary

In this chapter, we reviewed several influential NLI datasets, either expert-annotated or crowd-sourced, and pointed out the issues in dataset creation. We also discussed the symbolic and neural models that have been proposed for solving NLI. As one can see, there is a surge in research on NLI, manifest in the large number of new corpora, as well as the introduction of new logic-based models and neural network models. In fact, several recent studies experimented with hybrid systems which have both a symbolic and a neu-

ral component, for instance Hy-NLI (Kalouli et al., 2020) and NeuralLog (E. Chen et al., 2021).

However, there are still many unresolved problems. For instance, the current NLI datasets are not challenging enough for transformer models, and often exhibit annotation biases. Relatively few logic-based models have been proposed that exclusively focus on monotonicity. With neural models, too few resources exist for languages other than English, and claims on their cross-lingual abilities have not been thoroughly scrutinized. The next four chapters are in a way written against the backdrop of these issues reviewed above.

| language(s) | source | translated? | size | name: short description |
|---|---|---|---|---|
| 15 languages | Conneau et al. (2018b) | MT+HT | 7.5k each lang. | XNLI: dev + test sets are translated by humans; training sets are machine-translated (using Facebook's in-house MT system). |
| Chinese | chapter 5; H. Hu et al. (2020b) | no | 56k | OCNLI: collected from scratch using enhanced MNLI-procedures. |
| Chinese | chapter 6; H. Hu et al. (2021) | no | 34k | consists of four categories of adversarial and probing datasets |
| Portuguese | Fonseca et al. (2016) | no | 10k | ASSIN: sentences were retrieved from online news platforms and automatically matched based on a similarity measure to form NLI pairs. |
| Portuguese | Real et al. (2018) | MT | 10k | SICK-BR: machine translated SICK to Brazilian Portuguese; pairs were then manually checked by experts. |
| Portuguese | Real et al. (2020) | MT + synthesized | 10k | ASSIN2: based on SICK-BR, but included more entailment examples generated from SICK-BR. |
| Persian | Amirkhani et al. (2020) | no | 10k | FarsTail: created using multiple choice questions where correct answers are used for entailed hypothesis and incorrect answers are used for contradictions, similar to SciTail (Khot et al., 2018). |
| Japanese | Hayashibe (2020) | no | 48k | sentences from hotel reviews are extracted and automatically paired based on similarity or lexical overlap. |
| Turkish | Budur et al. (2020) | MT | 1,003k | SNLI-TR, MultiNLI-TR: machine-translated SNLI and MNLI into Turkish and asked humans to rate the translation quality (using Amazon's MT system). |
| Dutch | Wijnholds and Moortgat (2021) | MT | 10k | SICKNL: machine-translated SICK to Dutch (MT system not reported). |
| Hindi | Uppal et al. (2020) | no | 20.7k | recasting product reviews (and other existing datasets) to NLI: e.g., *has good streaming quality* ENTAIL *this product got positive reviews*. |
| Italian | Bos et al. (2009) | no | 800 | constructed on the basis of sentences from Wikipedia revision histories. |
| Arabic | Alabbas (2013) | no | 600 | sentences from new text were automatically extracted and paired, and then annotated by humans. |
| German | Eichler et al. (2014) | no | 24k | constructed from customer emails to the support center of a company and the category descriptions of the emails |

Table 2.10: Summary of NLI resources in languages other than English; MT: machine-translated, HT: human-translated.

# CHAPTER 3

# AUTOMATIC MONOTONICITY ANNOTATION

In this chapter, we discuss an algorithm and a tool that automatically annotate monotonicity arrows (also referred to as *polarities*) in a given input sentence.[1] This is the first step for making inferences with our proposed symbolic system; chapter 4 describes the second step.

Concretely, we want to determine the polarity ($^{\uparrow}$, $^{\downarrow}$ or $^{=}$) of all the words and constituents in a given sentence/text (as explained in chapter 2.2.1 and 3.2.2)

Once we have the polarities annotated, we can easily generate inferences using a replacement operation, and a knowledge base. For example, if we had a collection of background facts like *cats* ⌑ *animals*, *beagles* ⌑ *dogs*, *scares* ⌑ *startles*, and one ⌑ two, where we use ⌑ to denote the relation between the denotation of two words, then our $^{\uparrow}$ and $^{\downarrow}$ notations on *Every dog*$^{\downarrow}$ *scares*$^{\uparrow}$ *at least two*$^{\downarrow}$ *cats*$^{\uparrow}$ would allow us to conclude *Every beagle startles at least one animal*. More generally, $^{\uparrow}$ mandates that replacing the current word (*dog*) with another one of smaller extension (*beagle*) will result in an entailment, while for $^{\downarrow}$, the word should be replaced with another one of larger extension. This is the general recipe for obtaining monotonicity-related inferences.

The main goal of this chapter is to design and build a tool that performs this annotation automatically, for any input sentence, for example, sentences like *Every dog*$^{\downarrow}$ *scares*$^{\uparrow}$ *at least two*$^{\downarrow}$ *cats*$^{\uparrow}$, *Every dog*$^{\downarrow}$ *and no cat*$^{\downarrow}$ *sleeps* , and *Most rabbits hop*$^{\uparrow}$. Note that the $^{=}$ notation means that we can replace the word with neither a more specific term or a more general one. We call this task *monotonicity annotation* or *polarization*.

We achieve this by first expanding the van Benthem algorithm (van Benthem, 1986) to cover more rules in the Combinatory Categorial Grammar (CCG) formalism, and then per-

[1] This chapter is based on work presented in H. Hu and Moss (2018) and H. Hu and Moss (2020). The algorithm is designed in collaboration between me and my advisor Dr. Moss. I implemented it in Python and evaluated it on the evaluation set hand-crafted by me and Dr. Moss.

forming automatic annotation on parse trees obtained from freely available CCG parsers, for instance, C&C (S. Clark and Curran, 2007), EasyCCG (M. Lewis and Steedman, 2014) and depCCG (Yoshikawa et al., 2017). Our system is named ccg2mono because it takes in CCG parse trees and annotate monotonicity information on each node of the tree. It is released at https://github.com/huhailinguist/ccg2mono.

Using our polarization tool, we are able to make a very easy first step on automatic inference with only a syntactic tree representation, rather than logical forms that may be difficult to translate natural language into. Next chapter will discuss how this tool can be expanded to perform natural language inferences.

## 3.1 Research Questions

In this chapter, we aim to answer the following research questions:

1. How can the van Benthem algorithm (van Benthem, 1986) be extended to the full-fledged CCG for monotonicity tagging? That is, previous methods such as the van Benthem algorithm in polarization only consider the Ajdukiewicz/Bar-Hillel (AB) flavor of Categorial Grammar, where the rules are restricted to application rules ($>$) and ($<$), which are too restrictive to give wide-coverage grammars for natural language. What rules are needed to extend it for handling composition (B) and type-raising (T) rules in CCG?

2. How can we build and evaluate a monotonicity tagging system using available CCG parsers? We discuss the challenges we encounter on implementing a system that works on natural language input, notably the issues when using the CCG parse trees. We build the first hand-crafted evaluation dataset of polarized sentences and compare the performance of ccg2mono with the only existing tool NatLog.

3. What are the limitations of ccg2mono monotonicity tagging? We also discuss the limitations of ccg2mono and challenges arising from this work.

In the following, we will first introduce some preliminary knowledge, then describe and evaluate our monotonicity tagging system, and finally discuss the limitations of our current system.

## 3.2 Preliminaries

In order to automatically annotate monotonicity, we will first introduce preliminary concepts and theories, and then give a definition for monotonicity and polarity.

### 3.2.1 Syntax-semantics Interface from Combinatory Categorial Grammar

We adopt the categorial grammar framework (Carpenter, 1998; Keenan and Faltz, 1984), especially Combinatory Categorial Grammar (CCG) (Steedman, 2000), because there is a natural correspondence between syntax and semantics in CCG. In CCG, every word has a syntactic category (also called a supertag), and each syntactic category corresponds to a unique semantic type (a common noun $N$ corresponds to the type $\mathtt{p}e, t\mathtt{q}$). For example we have atomic categories such as *NP* and *S* which represent a noun phrase and a sentence respectively. There are also complex syntactic categories, for instance, *S*z*NP* which means it expects an *NP* to its left so that it will return a sentence. This is the category for a verb phrase which expects a subject noun phrase.

We give below two syntactic trees in the CCG formalism of the same sentence *cats chased dogs*. Note that in the first CCG tree we use the regular function applications, where in the second tree, we are intentionally making it more complicated by first type-raising (T) the subject *cats* and then use composition (B) in the next step. This is to demonstrate how type-raising and composition work. Readers are encouraged to consult Steedman, 2000

and other materials for a more detailed review of CCG.

$$\dfrac{\textit{cats} : np \qquad \dfrac{\textit{chased:}\ \mathtt{p}s\mathtt{z}np\mathtt{q}\{np \quad \textit{dogs} : np}{\textit{chased dogs} : s\mathtt{z}np}\ >}{\textit{cats chased dogs} : s}\ < \tag{3.1}$$

$$\dfrac{\dfrac{\textit{cats} : np}{\textit{cats:}\ s\{\mathtt{p}s\mathtt{z}np\mathtt{q}}\ \mathrm{T} \qquad \textit{chased:}\ \mathtt{p}s\mathtt{z}np\mathtt{q}\{np}{\textit{cats chased} : s\{np}\ \mathrm{B} \qquad \textit{dogs} : np}{\textit{cats chased dogs} : s}\ > \tag{3.2}$$

(3.2) is a syntactic tree in the CCG formalism. We need to "translate" it into a tree with semantic types, in order to apply the polarization algorithm, which is based on semantic principles. Below we translate syntactic types in (3.2) into semantic types, as shown in (3.3). We see the advantage of using CCG because we can easily obtain the semantics by using simple mapping rules such as: $S \rightsquigarrow t$ (a sentence is of type $t$), $N \rightsquigarrow e, t$ (a common noun is of type $e, t$), $NP \rightsquigarrow \mathtt{p}e, t\mathtt{q}, t$ (a noun phrase is of type $\mathtt{p}e, t\mathtt{q}, t$), etc.

$$\dfrac{\dfrac{\textit{cats:}\ \mathtt{p}e, t\mathtt{q}, t}{\textit{cats:}\ \mathtt{ppp}e, t\mathtt{q}, t\mathtt{q}, t\mathtt{q}, t}\ \mathrm{T} \qquad \textit{chased:}\ \mathtt{pp}e, t\mathtt{q}, t\mathtt{q}, \mathtt{ppp}e, t\mathtt{q}, t\mathtt{q}, t\mathtt{q}}{\textit{cats chased:}\ \mathtt{pp}e, t\mathtt{q}, t\mathtt{q}, t}\ \mathrm{B} \qquad \textit{Dogs:}\ \mathtt{p}e, t\mathtt{q}, t}{\textit{cats chased Dogs:}\ t}\ > \tag{3.3}$$

For a complete mapping from syntactic categories to semantic types, see Table 3.1.

| atomic syntactic type | semantic type | semantic type abbreviated |
|---|---|---|
| *n* | $e, t$ | N |
| *np* | $(e, t), t$ | NP |
| *s* | $t$ | S |
| *pp* | $pp$ | PP |

Table 3.1: Mapping from syntactic types to semantic types

### 3.2.2 Monotonicity and polarity

Before defining monotonicity and polarity[2], we will first introduce the idea of a preorder (Icard and Moss, 2014; Moss, 2012).

A *preorder* $\mathbb{P}$ $(P, \leq)$ is a set $P$ with a relation $\leq$ on $P$ which is reflexive and transitive. Reflexive means that for any $x$ in the set $P$, we have $x \leq x$. Transitive states that for $x$, $y$ and $z$ in $P$, if $x \leq y$ and $y \leq z$, then $x \leq z$. A simple preorder in algebra would be $(\mathbb{R}, \leq)$: the set of real numbers $\mathbb{R}$ and the usual "less-than-or-equal-to" ($\leq$) relation on the real numbers. The reason is that $\leq$ is both reflexive ($5.4 \leq 5.4$) and transitive (if $5.4 \leq 6.4$ and $6.4 \leq 7.4$, then $5.4 \leq 7.4$) in $\mathbb{R}$.

Moving on to natural language, we can define a preorder $\mathbb{P}$ $(\llbracket \textit{animals} \rrbracket, \leq)$, where $\llbracket \textit{animals} \rrbracket$ is the set of all animals and $\leq$ is the hypernym/hyponym relation ($poodle \leq dog$, meaning that *poodles* are a type of *dog*). Since animals are of type $e$ (entities), we can further note $\mathbb{P}$ as $\mathbb{P}_e$. We will also need a preorder for the truth values $\mathbb{P}_t$. We order this Boolean preorder $\mathbb{P}_t$ by $\mathbb{F} \quad \mathbb{T}$. For the higher types $x \to y$, we take the set $(\mathbb{P}_x \to \mathbb{P}_y)$ of all functions and endow it with the pointwise order. In this way every one of our semantic types is naturally endowed with the structure of a preorder in every model. For instance, even verbs which are interpreted as functions can be defined in a preorder: $\mathbb{Q}$ $(\llbracket \textit{moves} \rrbracket, \leq)$, where $\llbracket \textit{move} \rrbracket$ is a function from entity to truth values, which is of the type $e \to t$ (or written as $e, t$). We have $\leq$ relation which again is equivalent to hypernym/hyponym relation for such intransitive verbs: *sprint* $\leq$ *run*.

[2]This *polarity* is not to be confused with the negative or positive polarity in sentiment analysis.

Now we can give a definition of monotonicity. A function $f : \mathbb{P} \to \mathbb{Q}$ is *monotone* (or *order preserving*) if $p \sqsubseteq q$ in $\mathbb{P}$ implies $f(p) \sqsubseteq f(q)$ in $\mathbb{Q}$. And $f$ is *antitone* (or *order inverting*) if $p \sqsubseteq q$ in $\mathbb{P}$ implies $f(q) \sqsubseteq f(p)$ in $\mathbb{Q}$. For instance, a function $f(x) = 5x - 3$ is monotone because when $x$ increases, $f(x)$ also increases. A function can of course have more than one arguments, and then monotonicity needs to be defined over each argument: $g(x, y) = 5x - 6y$ is monotone in $x$ and antitone in $y$.

Another way of describing this would be using *polarity* on an argument. That is, if a function is monotone in one argument (say $x$), then that argument is said to have upward polarity, denoted by $^{\uparrow}$. If a function is antitone in one argument (say $y$), then $y$ will receive a downward polarity, $^{\downarrow}$. For this reason, *monotonicity* and *polarity* are sometimes used interchangeably in this dissertation. It is also important to point out that a function can be neither monotone nor antitone, for example, $z^2$ on $\mathbb{R}$. In this case, the polarity on $z$ will be

.

Our definitions above can be easily applied to natural language, since in CCG, interpreting a sentence is analogous in many cases to function application. To give an example in natural language, $[\![\textit{every}]\!] : (\mathbb{P}_{et}, \mathbb{Q}_{et}) \to 2$ is a function that takes two arguments: one of type $e, t$ in the preorder $\mathbb{P}$ (a noun, for example) and the other of type $e, t$ in the preorder $\mathbb{Q}$ (an intransitive verb, for example), and returns a Boolean (T or F). Since we know that *every* is antitone on the first argument and monotone on the second, we can re-write the interpretation of *every* as $[\![\textit{every}]\!] : (\mathbb{P}_{et}{}^{\downarrow}, \mathbb{Q}_{et}{}^{\uparrow}) \to 2$. Now, if we have 1) a sentence $S$: *every dog*$^{\downarrow}$ *swims*$^{\uparrow}$ (note we have already assigned the polarity of the words), and 2) a relation in $\mathbb{P}_{et}$: *poodle* $\sqsubseteq$ *dog*, then according to the definition of monotonicity, $S$: *every dog swims* strictly entails $S^1$: *every poodle swims*. The reason is that since the function of *every* is antitone on the first argument, and we have *poodle* $\sqsubseteq$ *dog* in the domain of the first argument, thus replacing *dog* with *poodle* in $S$ when the evaluation of $S$ is T, will result in a true value that is $\geq$ T. And the only truth value that is $\geq$ T is T itself. Thus when $S$ is T, we know that $S^1$ must also be T. Hence $S$ entails $S^1$.

### 3.2.3 Order-enriched types using *markings* ( , , and )

Monotonicity profiles of the quantifiers and other words have been discussed in the literature (Danescu et al., 2009; Dowty, 1994). That is, *every* is antitone on the first argument but monotone on the second argument; *no* is antitone on both arguments; *some* is monotone on both arugments; etc. Therefore, following Dowty (1994), we incorporate monotonicity information into the syntactic/semantic types. Function types $x$ Ñ $y$ split into three versions: the monotone version $x$ Ñ $y$, the antitone version $x$ Ñ $y$, and the full version $x$ Ñ $y$. (What we wrote before as $x$ Ñ $y$ is now $x$ Ñ $y$.) These are all preorders using the *pointwise order*. We must replace all of the ordinary slash types by versions of them which have *markings* on them, in order to give maximally informative annotations of the polarities.

In our system, we have three *markings* ( , , and ). Markings are used to tell if a function is interpreted (in every model) by a function which is always monotone ( ), always antitone ( ), or neither in general ( ). For example, *not* is an antitone function, which means it will flip the polarity; in other words, *not* is a downward entailing operator. Thus it will have the antitone marking ( ): p*NP* Ñ *S*q Ñ p*NP* Ñ *S*q (see next section for more examples).

### 3.2.4 Lexicon with order-enriched types

We use *S* for $t$, *N* or $et$ for $e$ Ñ $t$ $e$ Ñ $t$, *NP* for *N* Ñ $t$, *NP* for *N* Ñ $t$, and *NP* for *N* Ñ $t$. Note that we have a different font than our syntactic types $s$, $n$, and $np$. Also note that *N* can only be an argument to other functions, and cannot be a function itself. Thus thus markings on *N* is always , contrary to the markings on *NP* which can be any of the three markings. We will see why we need to do so in the next few sections. Then we use *NP* Ñ $S$ for intransitive verbs, *NP* or *NP* for noun phrases with determiners, $e$ for proper names. For the determiners, our lexicon then uses the order-enriched types in different ways:

| word | type | word | type |
|---|---|---|---|
| *every* | $N \tilde{\to} NP$ | *no* | $N \tilde{\to} NP$ |
| *some* | $N \tilde{\to} NP$ | *most* | $N \tilde{\to} NP$ |

For other words, we also have order-enriched types. It is especially important to have correct markings for words that flip the arrows, i.e., downward-entailing operators (DEOs), because the marking in their type is the only source for downward polarity in our system. They should all have one to be able to flip the polarity arrow. For example:

| word | type | word | type |
|---|---|---|---|
| *few* | $N \tilde{\to} NP$ | *refuse* | $(S \tilde{\to} NP) \tilde{\to} (S \tilde{\to} NP)$ |
| *not* | $(S \tilde{\to} NP) \tilde{\to} (S \tilde{\to} NP)$ | *if* | $(S \tilde{\to} S) \tilde{\to} S$ |
| *without* | $((S \tilde{\to} NP) \tilde{\to} (S \tilde{\to} NP)) \tilde{\to} NP$ | *rarely* | $(S \tilde{\to} NP) \tilde{\to} (S \tilde{\to} NP)$ |

Note that DEOs include not only quantifiers, and negators, but also prepositions, verbs, conditionals and adverbs. Several studies have designed supervised or unsupervised methods to identify these DEOs from raw text (Cheung and Penn, 2012; Danescu and L. Lee, 2010; Danescu et al., 2009). We use the words in their findings as our list of DEOs.

To define the markings on all words is an interesting and challenging task for lexical semantics, which is beyond the scope of this dissertation. In our current implementation, we hand-coded the well-known facts of determiners, downward-entailing operators, negators, etc. in the lexicon (see Table A.1 for an example). Additionally, to increase coverage, for all the words that do not receive a semantic type from the hand-coded lexicon, we give on all the arrows in their type by default, except for the arrows inside an NP. For instance, the "on" in the sentence "The man is carrying on" receives the semantic type: $(NP \tilde{\to} S) \tilde{\to} (NP \tilde{\to} S)$ from the parser, which is not covered in our hand-crafted lexicon. We will therefore assign the following markings to it $(NP \tilde{\to} S) \tilde{\to} (NP \tilde{\to} S)$. Note here that the markings on the two *NP*s are still unspecified (note that *NP* *NP* ), but all $\tilde{\to}$ are now $\tilde{\to}$.

To be more specific, only words in our hand-crafted lexicon, or nouns and pronouns come with markings on *NP*s; the markings on all other *NP*s start out as under-specified, and will be determined via propagation from *NP*s with markings (which happens in the `mark` phase of our algorithm).[3]

### 3.2.5 A note on notation

The ¤ in this dissertation is used for words and phrases, which is equivalent to MacCartney's forward entail relation „ and equivalence relation  (chapter 5 of MacCartney, 2009).

Sentence-level entailment is usually represented as „ in this dissertation, while † represents non-entailment (neutral and contradiction). See Chapter 1.1.1 for a review of different entailment relations proposed in the literature.

## 3.3 Polarizing a CCG Parse Tree

In this section, we describe the algorithm we developed to polarize a CCG parse tree. Our algorithm has two steps: *mark* and *polarize*. *Mark* will assign one of the three *markings* on all the semantic types so that they are enriched with the order information, from tree leaf to tree root. *Polarize* will then assign monotonicity arrows to all the constituents in the tree, from root to leaf.

**Input** A parse tree $\mathcal{T}$ in CCG as in (3.2), and a marked lexicon.

**Output** We aim to convert $\mathcal{T}$ to a different tree $\mathcal{T}$ satisfying the following properties: (1) The semantic terms in $\mathcal{T}$ and $\mathcal{T}$ should denote the same function in each model. (2) The lexical items in $\mathcal{T}$ must receive their types from the typed lexicon. (3) The polarity of the root of $\mathcal{T}$ must be Ò. (4) At each node in $\mathcal{T}$, one of the rules in our system must be matched.

[3] We are yet to formally prove the soundness of this implementation, but extensive experimentation on the evaluation set in chapter 3.5 and other hand-crafted challenging sentences has shown that results from this implementation are reliable.

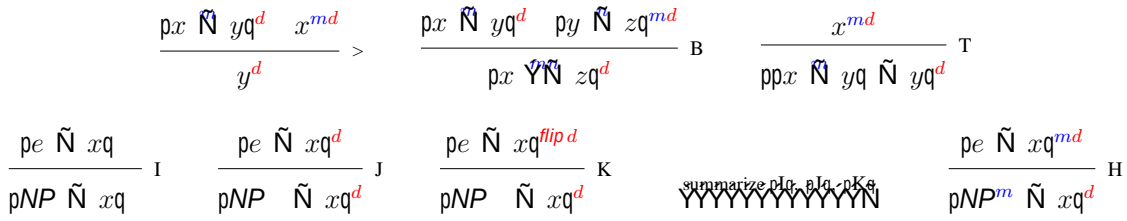
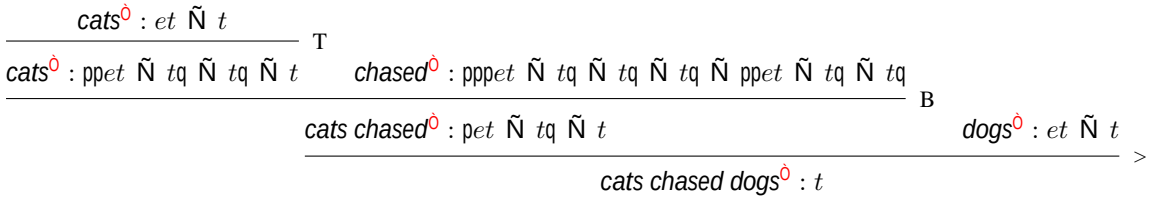


Figure 3.1: The top line contains core rules of marking and polarization. The letters $m$ and $n$ stand for one of the markings , , or ; $d$ stands for Ò or Ó (but not ). In (I), (J), (K) and (H), $x$ must be a boolean category. See charts in the text for the operations $m$ $d$ ÑÑ $md$ and $m$ $n$ ÑÑ $mn$. In line with H. Hu and Moss, 2018, we present (I), (J), (K) rules separately, but also show a summary in rule (H) which is important for computation of , for instance, when $d$ Ò, $m$ , then $d$ $m$ ÑÑ .

**Example output** For $\mathcal{T}$ in (3.2), $\mathcal{T}$ will be (3.4):

(3.4)



The rules by which we assign markings and polarities on each node of a CCG parse tree are presented in Figure 3.1. Note that each node in Figure 3.1 represents both the *mark* rules and the *polarize* rules, where the blue color shows marking rules, and the red color shows the polarization rules. Next, we will explain the *mark* rules and *polarize* rules separately.

### 3.3.1 Step 1: Mark

In this step, we start with a parsed CCG tree, and a pre-compiled lexicon with order-enriched types, i.e., with markings on all the words (leaf nodes) from the lexicon. The goal of *mark* is to propagate the markings from the leaf nodes to all the intermediate nodes in the parse tree, using the *mark* rules in Figure 3.1 that are summarized below in Figure 3.2:

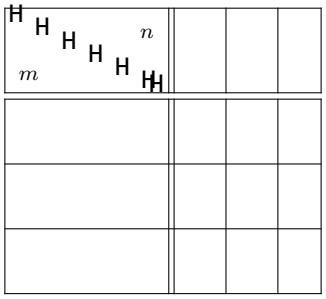


Figure 3.2: *mark* rules

The rules state that when two markings $m$ and $n$ are to be joined, if $m$ and $n$ are either or , then the result will be when $m = n$, and it will be when $m \neq n$. Whenever either $m$ or $n$ is , the result will always be . So far these rules are only effective in the composition rule (B) presented in Figure 3.1.

In most cases, the markings will just be inherited. For example, for the function application rule (>)

$$\frac{x \text{ Ñ } \text{p}y \text{ Ñ } z\text{q} \quad x}{y \text{ Ñ } z} > \tag{3.5}$$

will become

$$\frac{x \text{ Ñ } \text{p}y \text{ Ñ } z\text{q} \quad x}{y \text{ Ñ } z} > \tag{3.6}$$

after the *mark* phase. Note that here we are populating the tree leaf-to-root (i.e., from top to bottom).

### 3.3.2 Step 2: Polarize

In this step, we start with a marked CCG tree after the *mark* phase where all the leaf nodes and intermediate nodes have now order-enriched types, and assign arrows on them, from the root to the leaf, according to the polarizing rules.

The polarizing rules in Figure 3.1 are summarized below in Figure 3.3:

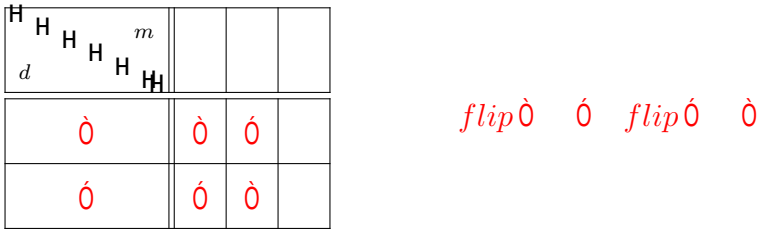



Figure 3.3: *polarize* rules

In Figure 3.1, $x$, $y$ and $z$ are variables ranging over marked types.

The application rule (>) is essentially taken from van Benthem (1986) (see also Lavalle-Martínez et al. (2017) for a survey of related algorithms).[4] All the other rules are new.

To illustrate the polarization rule for (>), let us take $m$ and $d$ Ò. We then start with the (>) rule

(3.7)
$$\frac{px \text{ Ñ } yq \quad x}{y^{Ò}} >$$

which will first become

(3.8)
$$\frac{px \text{ Ñ } yq^{Ò} \quad x}{y^{Ò}} >$$

That is, the functor will receive the same arrow from below, according to Figure 3.1.

Then, we compute the arrow on $x$, which will be: Ò Ó, according to Figure 3.3,

[4]Other polarization algorithms may also be used, as reviewed in Lavalle-Martínez et al. (2017). For this work, we experiment with van Benthem's algorithm (van Benthem, 1986) for its compatibility with the CCG formalism.

resulting in

$$\frac{\mathrm{p}x \;\tilde{\mathrm{N}}\; y\mathrm{q}^{\grave{\mathrm{O}}} \quad x^{\acute{\mathrm{O}}}}{y^{\grave{\mathrm{O}}}} >$$

(3.9)

Note that this polarization goes from root to leaf (bottom to top).

Recall that van Benthem (1986) only specified the rules for function application, but in CCG we have other rules such as composition (B) and type-raising (T), which are also important since our goal is to cover as much natural language as possible. We also need more rules such as (I), (J) and (K) to correctly polarize *NP*s with $no$ as the determiner. Thus in the next paragraphs we specify how *mark* and *polarization* work for these rules.

**Rules (I), (J), and (K)** The rules (I), (J), and (K) are novel. In them, $x$ must be *Boolean*. That is, it must belong to the smallest collection $B$ containing $t$ and with the property that if $z \;\mathrm{P}\; B$, then $\mathrm{p}y \;\tilde{\mathrm{N}}\; z\mathrm{q} \;\mathrm{P}\; B$ for all $y$. $B$ is thus the collection of types whose interpretations are naturally endowed with the structure of a *complete atomic boolean algebra* (Keenan and Faltz, 1984). Indeed, the soundness of (J) and (K) follows from the proof of the Justification Theorem (op. cit). Rule (H) is a summarization of (I), (J), and (K).

Figure 3.4 contains two applications of the (K) rules. First, the lexical entry for *chased* is $e \;\tilde{\mathrm{N}}\; et$. The first application of (K) promotes this to *NP* $\tilde{\mathrm{N}}\; et$. The *NP* receives a because its argument $no\,cat$ is of type *NP* . Note that the polarity flips when the (K) rule is applied. If we had used (J), the promotion would be to *NP* $\tilde{\mathrm{N}}\; et$, and there would be no polarity flipping, which would give us the wrong polarity. This would be used in a sentence where the object VP was *some cat* or *every cat*. The second application promoted *chased no cat* from the type $et$ to *NP* $\tilde{\mathrm{N}}\; S$, again with a polarity flip. If we had used (I), we would have obtained *NP* $\tilde{\mathrm{N}}\; S$. However, this would have trivialized the polarity to , and this effect would have been propagated up the tree. Rule (I) would be needed for the

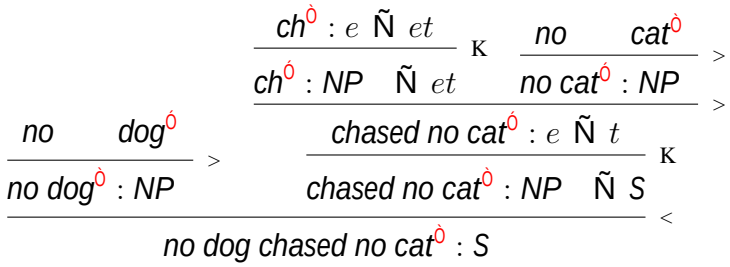


Figure 3.4: Two applications of the (K) rules.

sentence *most dogs chased no cat*.

**Backward rules** "Backward" versions of (>), (B), and (T) are similar to those in Figure 3.1. The important point is to make sure which is the functor and which is the argument for the function. For example, we show the forward function application (>) and its corresponding backward version (<) below. In both rules, $px$ $y$q is the functor while $x$ is the argument whose polarity is determined by $m$ $d$.

$$\frac{px\ y^{d}\quad x^{md}}{y^{d}}\ >\qquad\qquad \frac{x^{md}\quad px\ y^{d}}{y^{d}}\ <$$

**Boolean connectives** We take *and* and *or* to be polymorphic of the types $B_1$ pB$_2$ B$_3$q, when $B$ is a Boolean category and $m$ , , or .

Note that the markings on $B_1$ and $B_2$ are inherited from the conjuncts, but the marking of $B_3$ needs to be computed from $B_1$ and $B_2$. In the *mark* phase, if $B_1$ and $B_2$ have the same marking (for instance, , then $B_3$ will also be marked the same. However, it becomes challenging when the two conjuncts do not have the same marking, for instance, *some men and no women ran* , where *some men* is *NP* but *no women* is *NP* , as shown in the tree in Figure 3.5. For such cases, the type of $B_3$ will be , i.e., *and* : *NP* pNP *NP* q.

Also note the use of rule (H) in this example, which gives us the correct polarity on *ran*. This can be verified from the following non-entailment relations:

(1) a. *Some men and no women ran*? † *Some men and no women move*: is wrong

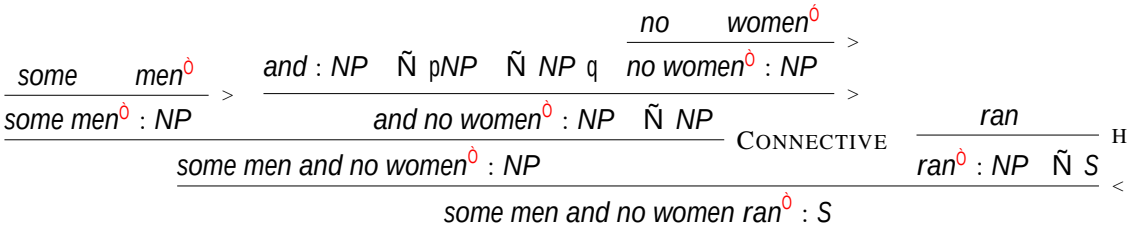



Figure 3.5: Tree involving a Boolean connective

for *ran*

b. *Some men and no women ran*? † *Some men and no women ran fast*: ó is wrong for *ran*

Consequently, a polarity should be assigned to *ran*.

**Other combinators** This chapter only discusses (T) and (B), but we also have rules for the other combinators used in CG, such as (S) and (W). For example, the (S) combinator is defined by $Sfg \quad \lambda x.\mathfrak{p}fx\mathfrak{q}\mathfrak{p}gx\mathfrak{q}$. In our system, the corresponding polarization rule is

$$\frac{\mathfrak{p}x\ \tilde{\mathrm{N}}\ \mathfrak{p}y\ \tilde{\mathrm{N}}\ z\mathfrak{q}\mathfrak{q}^{d} \quad \mathfrak{p}x\ \tilde{Y}\tilde{\mathrm{N}}\ y\mathfrak{q}^{nd}}{\mathfrak{p}x\ \tilde{\mathrm{N}}\ z\mathfrak{q}^{d}}\ \mathrm{S}$$

This combinator is part of the standard presentation of CCG, but it is less important in this chapter because the parsers do not use it.

### 3.3.3 An Example

Here we show an example for the sentence: *John refused to dance without pants*. Note that our intuitions give us a downward arrow on *dance*ó because the sentence entails that *John refused to waltz without pants*, but an upward arrow on *pants*ò because it also entails that *John refused to dance without clothes*.

*First*, we will need to obtain the CCG parse tree, and then the semantic types of the words (from a pre-defined lexicon) in the tree, as shown in Figure 3.6.

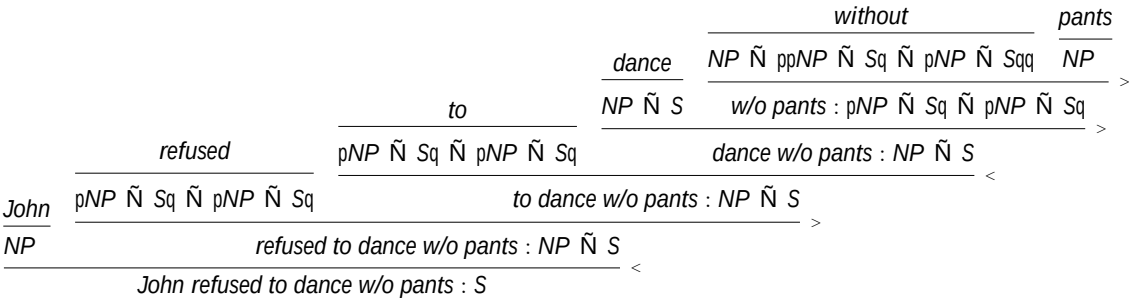



Figure 3.6: CCG tree after putting in the semantic types from our lexicon.

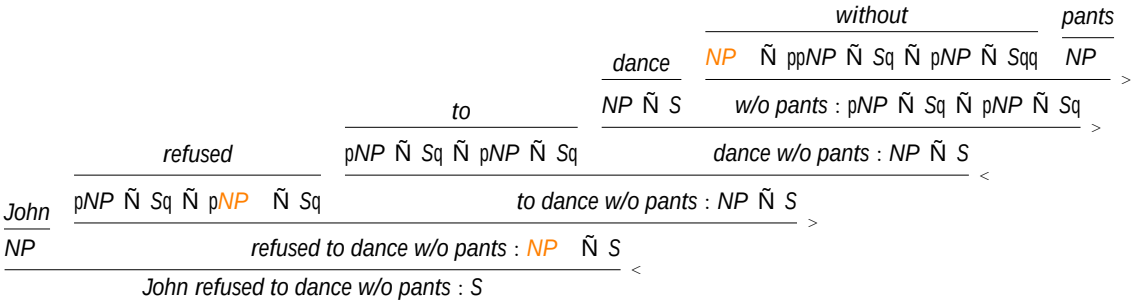



Figure 3.7: CCG tree after **marking**, *NP* shows where *NP* has been propagated.

There are several points to note in the tree in Figure 3.6. 1) Most of the *NP* in the types of the trees are unspecified at the moment. The only ones that are certain to be of *NP* are person names such as *John* and nouns such as *pants*. The rest of the *NP*s will be specified in the *mark* phase, based on *John* and *pants*. 2) For space reasons, we left out the LEX rules which turns the *N pants* into an *NP* . 3) Most of the types in the intermediate nodes are unspecified too, indicated by the dot .

*Second*, we show the tree after **mark** in Figure 3.7.

In the tree above, note that the *NP* (*John* and *pants*) is populated to the orange *NP*s. The Ñ in the intermediate nodes have also been marked. This is result of the **mark** phase. It is worth pointing out that the other *NP*s in the tree are still unspecified, and they will not affect the final results of the monotonicity arrows.

*Finally*, Figure 3.8 shows our tree after **polarize**, with monotonicity information tagged for every word and constituent:

Importantly, the two Ñ in *refused* and *without* serve to flip the monotonicity arrows, which ensures that we obtain the correct predictions on *dance* and *pants*.

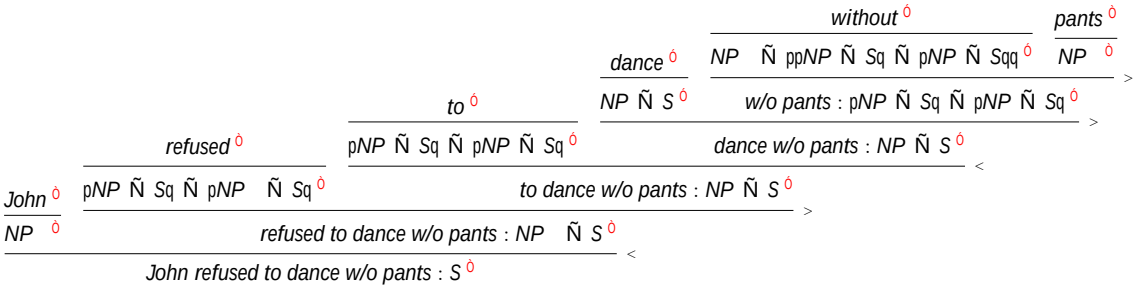



Figure 3.8: CCG tree after **polarization**.

## 3.4 Pre-processing and Post-processing Details

In pre-processing, we tokenize input sentences using the ccg2lambda system (Martínez-Gómez et al., 2016). The tokenized sentences are then parsed using a CCG parser (for instance the C&C parser (S. Clark and Curran, 2007) or the EasyCCG parser (M. Lewis and Steedman, 2014), which is trained on the CCGbank (Hockenmaier and Steedman, 2007) or the rebanked CCGbank (Honnibal et al., 2010)). Then we run our algorithm to mark and polarize the CCG tree.

Since our system relies on the CCG parses, it is essential to have correct CCG parse trees as input. We now describe some of the post-processing we do to systematically "correct" the parse trees to conform to our semantic representation. Note that our system will not be able to handle cases where the CCG parse trees contain too many errors.

It is also important to note that most of the post-processing has to do with training data for the parsers, i.e., the CCGbank (Hockenmaier and Steedman, 2007), rather than a particular parser alone. Some of the issues can be alleviated by using a model trained on the rebanked CCGbank (Honnibal et al., 2010), as the relative clauses in the rebanked CCGbank has the desired structure our system expects (see below). Both `EasyCCG` and `depCCG` provide models trained with the rebanked CCGbank.

**Correcting *most*** First, the parser assigns the supertag `N/N` to *most*, but `NP/N` to other quantifiers. Thus in order to handle *most*, we manually change the supertag of *most* and adjust the part of the tree structure.

**Relative clauses** Parsers trained on the original CCGbank parse relative clauses as *(no dog) (who chased a cat) died* rather than *(no (dog who chased a cat)) died*. Our system corrects the relative clauses into the latter structure.

**Other errors** Furthermore, the parsers sometimes behave differently on intransitive verbs likes *walks* than on *cries*. At the moment, we do not have a good strategy for correcting these errors. Our system has to accept the parse trees from the parser in this case.

Finally, we are working with the most probable parse tree returned by the parser, so sentences with scope ambiguities will be interpreted as-is from the tree returned by the parser, without any further processing such as type-raising and quantifier raising.

## 3.5 Evaluation of `ccg2mono`

In order to evaluate our system, we created a small evaluation dataset where sentences are hand-annotated for the polarity arrows. To the best of our knowledge, though small, this is the first evaluation dataset for monotonicity marking.

### 3.5.1 Creating an Evaluation Dataset

We hand-crafted 56 sentences containing a wide range of quantifiers, mixed with conditionals and conjunctions (see below) to test the system's polarization ability (see Table 3.2). These sentences are chosen to cover a wide range of monotonicity related linguistic phenomena, with some of them taken from examples sentences in Steinert-Threlkeld and Szymanik (2019). We manually annotated the polarity/monotonicity labels for each token. The annotation was performed by myself, in consultation with my advisor Dr. Larry Moss who is an expert in monotonicity in language and logic. We believe the monotonicity annotation requires expert knowledge and is thus very difficult for crowd-workers to annotate.

| Sentence | Linguistic phenomenon |
|---|---|
| Some↓ rat↓ sees↓ every↓ squirrel↓ | some/every |
| Most↓ dogs chase↓ some↓ cat↓ | most/some |
| Many↓ people like↓ dogs↓ as↓ pets↓ | many |
| At↓ least↓ seven↑ fish↓ died↓ yesterday↓ in↓ Morocco↓ | at least n |
| My↓ parents↓ said↓ I↓ could↓ have↓ three↑ candies↑ | numbers |
| Three out↑ of↑ five dentists recommend↓ that↓ their↑ patients↑ brush↓ their↑ teeth↓ at↓ least↓ four↑ times↓ a↓ day↓ | numbers |
| If↓ every↑ cat↓ runs↑ ,↓ then↓ some↓ dog↓ runs↓ also↓ | conditional |
| A↓ dog↓ who↓ ate↓ two↑ rotten↑ biscuits↑ was↓ sick↓ for↓ three↑ days↑ | relative clause/numbers |
| Ursula↓ refused↓ to↓ sing↑ or↑ dance↑ | disjunction |

Table 3.2: Example sentences in our evaluation dataset, with hand-annotated monotonicity information.

### 3.5.2 Evaluation Setup

We tested our system, as well as the `NatLog` system in MacCartney, 2009[5] on the small evaluation dataset and report results for several metrics.[6] We chose `NatLog` because it is the only system that performs polarization at the time of writing. As a reminder, `NatLog` uses dependency parse trees and heuristics of the important operators for monotonicity (negations, quantifiers) to perform polarization.

We group the words into two categories.

- *Key words*: content words (nouns, verbs, adverbs, adjectives), determiners, and numbers
- *Other words*: all words with other part-of-speech tags, for instance, prepositions.

The *key words* are important for making inferences in downstream tasks, as they can be replaced by words from the knowledge base (or preorders) to generate entailed statements.

[5] We used the `natlog` annotator in Stanford CoreNLP 3.9.2: https://stanfordnlp.github.io/CoreNLP/natlog.html. See a description of `NatLog` in chapter 2.2.1.

[6] As this is a completely new task in NLP, there is no prior work to consult with. We chose these metrics based on our work in monotonicity annotation, as well as their significance in downstream inference tasks.

The NatLog system uses dependency parses and heuristic rules (for instance, *without* will flip the arrow of its dependent) to annotate Ò and Ó, but not .

### 3.5.3 Evaluation Results

First, out of the 56 sentences, the EasyCCG parser (M. Lewis and Steedman, 2014) produced parse trees that contain errors for 11 sentences. The NatLog system depends on dependency parses given by the Stanford dependency parser, which produces 9 wrong parses. Since both systems make use of parse trees, both are in general unlikely to correctly annotate polarity if the parse tree is problematic.

Then, we evaluate the accuracy of polarity annotation on both the token level and the sentence level, following the evaluation procedures for part-of-speech tagging evaluation (Christopher D Manning, 2011). We performed one evaluation on all tokens, another only on the *Key words* described above; the rationale is that it is hard to determine what the correct polarity should be assigned to function words such as the *as* in *Many*Ò *people like*Ò *dogs*Ò *as*Ò *pets*Ò. Most of the useful polarity information for inference is on content words.

The results are shown in Table 3.3. ccg2mono outperforms NatLog at both the token-level (for key words, 69.4% vs. 78.2%) and the sentence-level (for key words, 50.0% vs. 28.6%). Both systems perform much better than the majority baseline which assigns Ò to every token (on token-level, both accuracies from the two systems are 20-30% better than the baseline). Now we look at the mistakes of the two systems. NatLog considers "many" and "most" to be non-monotonic on both the first and second argument; it also treats "the" as "a" in that both arguments of "the" receive Ò polarity (*the dog*Ò *is chasing the cat*Ò), which can be controversial because as *the* is highly context-dependent, it is not entirely clear whether *the dog* will always entail *the animal* (also see discussion in chapter 3.6.2). Negation is sometimes handled incorrectly in NatLog where the NP in "NP doesn't VP" is tagged Ó. Both systems are unable to disambiguate the universal and existential "any", with NatLog tagging all "any" as a universal "any" and our system tagging

| | token-level | | | sentence-level | | |
|---|---|---|---|---|---|---|
| system | majority | NatLog | ccg2mono | majority | NatLog | ccg2mono |
| accuracy (all words) | 51.0 | 70.8 | 76.0 | 5.4 | 28.0 | 44.6 |
| accuracy (key words) | 49.1 | 69.4 | 78.2 | 5.4 | 28.6 | 50.0 |

Table 3.3: Accuracy (%) of NatLog and ccg2mono on the small evaluation dataset for polarity tagging.

it as the existential "any". Our system also finds it hard to recognize multi-word expressions such as "except for". While we can correctly tag conditionals and complements of verbal downward entailing operators (for example, "refuse") as ó, NatLog is incapable of handling such cases correctly. Overall, our system outperforms NatLog on this small evaluation dataset.

We intentionally put some hard examples involving numbers into the evaluation set, for instance, the second to last example in Table 3.2. The difficulty lies in determining whether the numbers are to be interpreted as *at least n*, *at most n* or *exactly n*. The correct polarities often cannot be determined without an understanding of the context and some world knowledge.

## 3.6 Discussion and Challenges

### 3.6.1 Properties of Our Algorithm

**Soundness** We have proved a *soundness theorem* for the system. As this is not the focus of the dissertation, we refer interested readers to S. Moss and H. Hu (in prep).

**Completeness** We have not proven the completeness of our system/algorithm, and indeed this is an open question. What completeness would mean for a system like ours is that whenever we have an input CCG parse tree and a polarization of its words which is semantically valid in the sense that it holds no matter how the nouns, verbs, etc. are interpreted, then our algorithm would detect this. This completeness would be a property of the rules

and also of the polarization algorithm. The experience with similar matters in Icard and Moss (2013) suggests that completeness will be difficult.

**Efficiency of our algorithm** Our polarization is fast on the sentences which we have experimented it on. We conjecture that it is in polynomial time, but the most obvious complexity upper bound to the polarization problem is NP. The reason that the complexity is not "obviously polynomial" is that for each of the type raising steps in the input tree, one has three choices of the raise. In more detail, suppose that the input tree contains

$$\frac{x}{\mathtt{p}x \mathbin{\tilde{\mathtt{N}}} y\mathtt{q} \mathbin{\tilde{\mathtt{N}}} y}\ \mathrm{T}$$

Then our three choices for marking are: $\mathtt{p}x \mathbin{\tilde{\mathtt{N}}} y\mathtt{q} \mathbin{\tilde{\mathtt{N}}} y$, $\mathtt{p}x \mathbin{\tilde{\mathtt{N}}} y\mathtt{q} \mathbin{\tilde{\mathtt{N}}} y$, and $\mathtt{p}x \mathbin{\tilde{\mathtt{N}}} y\mathtt{q} \mathbin{\tilde{\mathtt{N}}} y$. Our implementation defers the choice until more of the tree is marked. But prima facie, there are an exponential number of choices. All of these remarks also apply to the applications of (I), (J), and (K); these do not occur in the input tree, and the algorithm must make a choice somehow for these three rules.

### 3.6.2 Challenges for Monotonicity Tagging

While the enrichment of the types of most words are rather straightforward, there are several word classes that still remain challenging.

**Challenge 1: prepositions** One example is the class of prepositions. It is clear that prepositions such as *without* are downward-entailment operators:[7]

(2) He went to the exam **without** reading the book[6] „ He went to the exam without reading the chapter.

However, for other prepositions, it can become challenging:

[7] We use $\sqsubseteq$ to denote entailment, and $\not\sqsubseteq$ for non-entailment. $\sqsubseteq$ essentially covers both the equivalence and strict forward entailment relations in MacCartney's taxonomy (see Table 1.1 in chapter 1.1.1).

(3) a. He lives **in** Paris „ He lives in Europe

b. He likes living **in** Paris {? † He likes living in Europe

In (3a), *in* is upward entailing, but in (3b), it is unclear what role *in* plays. Furthermore, the many and widespread idiomatic uses/fixed phrases of prepositions make it even more difficult to assign the order sign to the types:

(4) a. **in** short/time/general/...

b. **on** leave/vacation/duty/...

In short, prepositions are extremely complicated in English and their different uses will require more careful investigation.

**Challenge 2: *the*** As the most frequent word in English, *the* poses unique challenge to our task.

(5) The small chair is in the bedroom. „ ? The chair is in the bedroom.

(6) The small chair is in the bedroom. „ ? The small chair is in the room.

Whether the above inferences hold seems to rely heavily on what exactly the chair and the room are referring to. It is likely that *the chair* and *the small chair* refer to the same entity, if there is only one chair in the context. However, we do not have the same intuition for *the room* and *the bedroom*, perhaps because usually we do not call the bedroom *the room*, since there are most likely other rooms such as the living room in the context.

Such questions related to the definite articles are beyond the scope of this dissertation. In the current implementation of `ccg2mono`, *the* is assumed to have the type: $N$ Ñ *NP* .

**Challenge 3: *any*** The determiner *any* can be either a free-choice item (FCI) or an negative polarity item (NPI) (Chierchia, 2013). As an FCI, it behaves like *every* in terms of monotonicity, as shown in (7a), where the first argument of *any* has downward monotonicity, i.e., *any* flips the arrow. However, when *any* behaves as an NPI, it does not flip the

arrow of its argument, as in (7b); note that the downward arrow is a result of the negation *not*, rather than *any* in this example.

(7) a. FCI: **Any** high school student$^{\acute{O}}$ can solve this problem. „ Any female high school student can solve this problem.

b. NPI: John has **not** tasted **any** cookies$^{\acute{O}}$ in this box. „ John has not tasted any mint-flavored cookies in this box.

Thus, to correctly predict the monotonicity of the arguments of *any*, we need to first disambiguate every *any*, which is beyond the scope of this dissertation.

**Challenge 4: numbers** Numbers pose a major challenge to us. Here we present two examples:

(8) a. John ate three $^{\{?}$ cookies.

b. John can finish three$^{\acute{O}}$ hamburgers in five$^{\grave{O}}$ seconds. (in a eating competition)

c. John can finish the course with three$^{\grave{O}}$ strokes in ten$^{\grave{O}}$ minutes. (in a golf game)

In (8a), it seems reasonable to assume that *three* receives =, but if *three* is interpred as *at least three*, then this sentence entails *John ate two cookies*. In (8b), this sentence entails that he can eat two hamburgers in 20 seconds. However, in the case of a golf game, where the fewer strokes the better, we will have an upward arrow on the *three*, since if John can do it in three strokes, he should be able to do the same within four, five or even more strokes. This shows that, the inferences depend heavily on the context and expectation of the sentence.

There is a long line of research both in theoretical linguistics (Horn Laurence, 1972; Kennedy, 2013) and language acquisition (Huang et al., 2013; Musolino, 2004, 2009) on the monotonicity and inferences for numbers. It seems almost impossible to encode such monotonicity information in a purely semantic system like `ccg2mono`. It will be an in-

teresting line of research to collect more empirical data and model them using models for pragmatics.

**Challenge 5: generics** In English, sometimes plural nouns can serve as a full noun phrase and it is often difficult to determine what these generic noun phrases mean.

(9) a. John is cooking tomatoes$^{\grave{0}}$. „ John is cooking vegetables.

b. Chairs$^{\acute{0}}$ are intended for sitting, not standing. „ Small chairs are intended for sitting, not standing.

The intended meaning of 'tomatoes' in (9a) is 'some tomatoes', where in (9b), 'chairs' most likely means 'all chairs'. There seems to be some correlation between the syntactic position of the generics (subject vs object) and whether it should be interpreted as an existential or a universal quantifier. This is another interesting line of research to pursue.

Finally, as mentioned in chapter 3.4, errors from the CCG parsers will likely affect the results of polarization. Since current CCG parsers are all trained on the CCGbank which only contains text from the Wall Street Journal, it is potentially helpful to make use of parsers that cover a wider range of genres, albeit using a different grammar formalism, such as the Udep2Mono system (Zeming Chen and Qiyue Gao, 2021) which uses the binarized Universal Dependency grammar.

## 3.7 Summary

In this chapter, we presented an extension of the van Benthem algorithm (van Benthem, 1986) that can handle polarization involving all the CCG rules, and an implementation of the algorithm that takes in CCG parse trees and annotates the monotonicity information on every constituent.

I have implemented our algorithm in Python. This implementation can read CCG parses from three parsers: `C&C` (S. Clark and Curran, 2007), `EasyCCG` (M. Lewis and Steedman, 2014) and `depCCG` (Yoshikawa et al., 2017). Implementing a system that works directly

on any natural language input is a non-trivial step on top of the theoretical advance because the parses delivered by the parsers may deviate in several respects from the semantically-oriented analyses that semanticists would prefer; thus we need to systematically correct the parser analyses (see chapter 3.4).

I also designed the first evaluation dataset for this task and showed that our system `ccg2mono` outperforms the dependency-based system `NatLog`.

Finally, we discussed several challenges in monotonicity annotation including prepositions, the article 'the', the quantifier 'any', numbers, and generic plural nouns, which point to interesting future directions of work in this field.

# CHAPTER 4

# INFERENCE WITH MONOTONICITY AND NATURAL LOGIC

The ultimate goal of having monotonicity information, described in the previous chapter, is to make sound inferences.[1] In this chapter, we introduce an inference engine called `MonaLog` that is based on `Mo`notonicity and `na`tural logic, and can be used as an inference engine to solve natural-logic-related inference problems.

After laying out the research questions, we will introduce background information on pre-orders and the knowledge base. Next, we will discuss the details of the `MonaLog` inference engine.[2] Finally I will present the results of testing `MonaLog` on two NLI datasets (FraCaS and SICK) and discuss some challenges.

## 4.1 Research Questions and Significance

In this chapter, we ask the following questions:

1. How to build a rule-based, symbolic inference engine that leverages the monotonicity information obtained from `ccg2mono`? Specifically, how to build a knowledge base and perform the replacement based on the monotonicity arrows?

2. How to incorporate natural logic rules into the inference engine?

3. How well can our system perform on existing NLI datasets? How does its performance compare with machine/deep learning models, notably the pretrained transformer models such as BERT (Devlin et al., 2019)?

To answer the first and second research questions, we designed and built an inference system called `MonaLog`, which is light-weight, relying only on syntactic tree structures,

[1] This chapter is based on H. Hu et al. (2019) and H. Hu et al. (2020a). I designed the `MonaLog` system in consultation with other authors on the two papers. I then implemented the system and ran all the experiments.

[2] https://github.com/huhailinguist/monalog

without the need to translate input sentences to logical forms. It handles well monotonicity-related inferences, and can be expanded by incorporating natural logic rules. To preview the results, evaluation on the first section of FraCaS shows that MonaLog performs competitively with previous logic-based systems. (Because of the small size of FraCaS, machine learning and deep learning models are rarely tested on FraCaS.) When evaluated on the SICK dataset, MonaLog performs competitively with other logic-based models, but falls several points behind deep learning models such as BERT. Additionally, MonaLog is, to the best of my knowledge, the first one that *generates* inferences based on natural logic, which can then be used as augmented training data for machine learning models.

Our results show that a monotonicity-based inference system is capable of achieving good performance on commonly tested NLI datasets for logic-based systems, suggesting that monotonicity related inference is wide-spread in these benchmarks. Apart from NatLog (MacCartney, 2009), MonaLog is the only inference system that is based entirely on natural logic, providing a competitive alternative for other more sophisticated logic-based systems.

### 4.2 Preliminaries on Pre-orders and Monotonicity

To see how MonaLog works, it is important to understand what a pre-order is as it is the foundation of monotonicity reasoning. Specifically, when we say "replacing a word with another denoting a smaller set in case of the Ó to obtain an entailed sentence" (for instance, *every dog*Ó *runs* entails *every poodle*Ó *runs*), the order on which we perform the replacement is a pre-order. That is *poodle* ¤ *dog* is a relation in the pre-order.

Concretely, a pre-order is a pair of a set $S$ and the relation ¤ over itself. The key point is that the relation ¤ is *reflexive* ($s$ ¤ $s$ for all $s$ P $S$) and *transitive* (if $r$ ¤ $s$ and $s$ ¤ $t$, then we have $r$ ¤ $t$) (Icard and Moss, 2014; Moss, 2012).

For our purposes, monotonicity is defined on functions whose domains are the pre-

orders. We say:

$$f(x) \text{ is monotone if } s \leq t \rightarrow f(s) \leq f(t) \tag{4.1}$$

$$f(x) \text{ is antitone if } s \leq t \rightarrow f(t) \leq f(s) \tag{4.2}$$

Note that for $f(x)$, $x$ and $f(x)$ can be in two different domains where each has a separate $\leq$ relation. However, for simple functions mathematics, we are defining the $\leq$ as the regular "less than or equal to" ($\leq$) relation on $\mathbb{R}$, for both $x$ and $f(x)$. In natural language, the set for $x$ might be entities and they have one type of $\leq$ relation, while the domain for $f(x)$ is $(t, f)$, which has a different $\leq$ relation (namely $f \leq t$).

A note on the terminology is in order. In order theory, which is what we are working with now, the terms "monotone" and "antitone" are used to describe the order in functions defined over pre-orders. However, in calculus, "monotonically increasing" and "monotonically decreasing" are often used to talk about the functions on $\mathbb{R}$, for instance $f(x) = 2x$, as we see in the next paragraph. Underlyingly, the definition of monotonicity is similar in the two fields.

To give some examples, the functions $f(x) = 2x$, $f(x) = sin(x)$ *when* $x \in [0, \frac{\pi}{2}]$ are monotone (monotonically increasing), because in the specified domain, when $x$ increases, $f(x)$ also increases; while $f(x) = 3x$, $f(x) = cos(x)$ *when* $x \in [0, \pi]$ are antitone (monotonically decreasing). Note that a function may have two arguments, for example, $g(x, y) = 5x - 3y$; in this case, $g(x, y)$ is monotone on $x$ but antitone on $y$ because when $x$ increases, $g(x, y)$ also increases, but when $y$ increases, $g(x, y)$ will decrease. Finally, there is another type of function that is neither monotone or antitone, for example, $h(x) = x^2$ for $x \in \mathbb{R}$, because it is in fact antitone on $[-\inf, 0]$ and monotone on $[0, \inf]$. In all these cases, the real numbers $\mathbb{R}$ and the relation $\leq$ together form a pre-order.

Now we show how monotonicity can be applied to natural language. We will use the word *every* as an example. Recall from chapter 3.2.2 that $[\![\textit{every}]\!] : (\mathbb{P}_{et}, \mathbb{Q}_{et}) \rightarrow 2$, where

$\mathbb{P}_{et}$ and $\mathbb{Q}_{et}$ are both preorders. [[*every*]] will return T if and only if for all $x$ P $D_e$, if $x$ P $\mathbb{P}_{et}$, then $x$ P $\mathbb{Q}_{et}$ (note that preorders are sets). For example, $\mathbb{P}_{et}$ could be [[*dog*]], and $\mathbb{Q}_{et}$ could be [[*swim*]]. Then [[*every dog swims*]] will return T if and only if all entities that are dogs also swim. Next we will show that the function [[*every*]] is antitone on its first argument $\mathbb{P}_{et}$ but monotone on its second argument $\mathbb{Q}_{et}$.

For this, we define the ¤ relation on the preorder $\mathbb{P}$ and $\mathbb{Q}$ to be the hypernym/hyponym relations in WordNet (Miller, 1995). Thus for $\mathbb{P}$ we have relations such as [[*poodle*]] ¤ [[*dog*]], for $\mathbb{Q}$ we have relations such as [[*swim*]] ¤ [[*move*]]. This can be informally understood as the superset/subset relations. That is, all poodles are dogs, and thus the set of poodles is a subset of dogs. Similarly, the set of entities that are swimming is a subset of the set of entities that are moving.

To show that [[*every*]] is antitone on its first argument $\mathbb{P}_{et}$, we need to show that (according to definition of antitonicity in (4.1)):

$$\text{if } p_1 \text{ ¤ } p_2, \text{then } [\![\textit{every}]\!](p_1)(q) \text{ ¥ } [\![\textit{every}]\!](p_2)(q) \tag{4.3}$$

Since F ¤ T, we know that when $[\![\textit{every}]\!](p_2)(q)$ F, (4.3) is always true. Thus, to prove (4.3) we need to show that:

$$\text{if } [\![\textit{every}]\!](p_2)(q) \text{ T, then } [\![\textit{every}]\!](p_1)(q) \text{ T} \tag{4.4}$$

This is relatively straightforward, because from $p_1$ ¤ $p_2$, we know that every $p_1$ is $p_2$, then from (4.4) we know that every $p_2$ is $q$, and therefore according to transitivity, every $p_1$ is $q$.

Similarly, we can show that [[*every*]] is monotone on its second argument. This is to say that by just using the monotonicity facts about quantifiers such as "every" and preorder relations such as *poodle* ¤ *dog*, we can already obtain many interesting inferences.[3]

[3]For more examples and illustrations, please see Icard and Moss (2014).

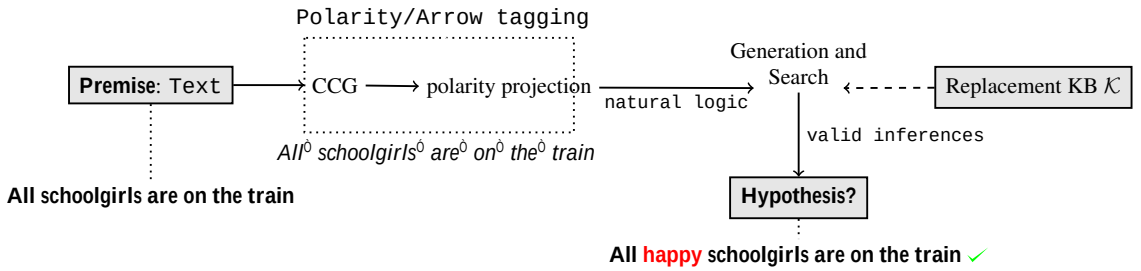



Figure 4.1: An illustration of our general monotonicity reasoning pipeline using an example premise and hypothesis pair: *All schoolgirls are on the train* and *All happy schoolgirls are on the train*.

Now we see that the preorder relations are central to making correct inferences on monotonicity. In the examples above, we see the usual ¤ relation for R in mathematics, and for natural language, the intuitive way to understand and define ¤ is the superset/subset relations such as *poodle* ¤ *dog*.

In MonaLog, the bulk of the knowledge base $\mathcal{K}$ has the ¤ relations as we have seen above, but it also contain some exclusion relations as in MacCartney's system (MacCartney, 2009). The details will be explained in chapter 4.3.2.

## 4.3 The MonaLog System

The goal of MonaLog is, for a given pair of *premise* and *hypothesis*, to determine their inference relation in the direction of *premise* Ñ *hypothesis*, which should be one of the following: entailment, neutral, and contradiction.

I present the pipeline of MonaLog in Figure 4.1, which involves three major steps: 1) Polarity/Arrow tagging; 2) Generation based on knowledge base $\mathcal{K}$; and 3) Search.

Specifically, given a premise text, we first do Arrow Tagging by assigning polarity annotations (ò, ó, ) to all tokens in the text, using ccg2mono as described in the previous chapter. These *surface-level* annotations, in turn, are associated with a set of natural logic inference rules that provide instructions for how to generate entailments and contradictions by span replacements over these arrows (which relies on a library of span re-

placement rules). For example, in the sentence *All schoolgirls are on the train* in Figure 4.1, the token *schoolgirls* is associated with a polarity annotation ↑, which indicates that in this sentential context, the span *schoolgirls* can be replaced with a semantically more specific concept (for example, *happy schoolgirls*) from the knowledge base $\mathcal{K}$ in order to generate an entailment. A `generation and search` procedure is then applied to see if the hypothesis text can be generated from the premise using these inference rules. A *proof* in this model is finally a particular sequence of edits (see Figure 4.2) that derive the hypothesis text from the premise text rules and yield an entailment or contradiction.

In the following sections, we provide the details of the different components and steps for `MonaLog`.

### 4.3.1 Polarization (Arrow Tagging)

Given an input premise $P$, `MonaLog` first polarizes each of its tokens and constituents, calling the `ccg2mono` system from chapter 3, which performs polarization on a CCG parse tree. For example, a polarized $P$ could be *every*$^{\downarrow}$ *linguist*$^{\uparrow}$ *swim*$^{\downarrow}$. Note that since we compare the generated inference with the input hypothesis on a strict string-to-string basis, we need to ignore morphology in the system; thus the tokens are all represented by lemmas.

### 4.3.2 Sentence Base $\mathcal{S}$ and Knowledge Base $\mathcal{K}$

`MonaLog` utilizes two auxiliary sets. The first is a sentence base $\mathcal{S}$, which stores the (uniquely) generated entailments, neutrals and contradictions in $\mathcal{S}$.entailments, $\mathcal{S}$.neutrals and $\mathcal{S}$.contradictions respectively.

Then we have a knowledge base $\mathcal{K}$ that stores all the relations needed for making inferences (see Table 4.1).

Now we will present different components of the relations in the $\mathcal{K}$. Note that the $\mathcal{K}$ can always be expanded manually, allowing new relations or vocabulary to be added, for

instance, wug ¤ thing, selfie ¤ photo, etc.

**Knowledge from WordNet (Miller, 1995)** WordNet is a thesaurus or ontology that contains more than 170,000 synsets, from which we obtain a large number of preorder relations, in the form of hypernym/hyponym. That is, for a word $w$, $w$'s hyponym ¤ word $w$ ¤ $w$'s hypernym. It has a good coverage for nouns (dog ¤ animal, dog | cat) and verbs. However, it does not include prepositions or other function words, which are required for some NLI problems. For instance, to determine that *turn on* is opposite of *turn off*, we need to have in the $\mathcal{K}$: on K off. While solving specific datasets such as FraCaS and SICK, we will manually encode lexical semantic relations that are needed into the $\mathcal{K}$. Note that we do not extract all possible hypernym/hyponym relations from WordNet. On the contrary, we extract these relations "on the fly" based on the vocabulary of the particular NLI problem the system is solving. That is, only if both the word *dog* and *animal* are in the premise-hypothesis pair do we add the dog ¤ animal from WordNet to our knowledge base.

**Knowledge about quantifiers** The knowledge base also contains relations among the quantifiers. This is important for obtaining entailment relation between *several dogs are barking* and *some dogs are barking*. We add the following relations to $\mathcal{K}$:

- *every all each* ¤ *most* ¤ *many* ¤ *a few several* ¤ *some a*; *the* ¤ *some a*. Note that these ¤ relations mean that if *every dog barks*, then *some dog barks*. Also, the treatment of the definite article *the* needs to be tailored to specific datasets. For instance, the SICK dataset considers *the* to be equivalent to *a*, and thus our $\mathcal{K}$ will be modified to have *the some a*. Here we treat *some* as the existential quantifier, rather than the reading of *several*.
- *at least/most* $n$.

  Because the CCG parsers do not treat *at least/most* $n$ as quantifiers as we hoped, we need a separate work-around for them. We first replace them with other quantifiers recognizable to the parser before parsing. After getting the parses, we put these

phrases back manually, along with the correct polarities. Specifically, we replace "at least *n*" with "some", "at most *n*" with "no" because they give the same polarities on their two arguments[4].

**Modifier relations** Semantic relations on individual lexical items as illustrated above are only the first step. To account for more phenomena in naturally occurring text, we incorporated the following ¤ relations:

- *modifier* + *noun* ¤ *noun*:
  this includes *adj* + *noun*, *noun* + *relative clause*, *noun* + *prepositional phrase* ¤ *noun*
- *modifier* + *verb* ¤ *verb*:
  this includes *adv* + *verb*, *verb* + *adv*, *verb* + *prepositional phrase* ¤ *verb*
- *degree adv* + *adj* ¤ *adj*: for instance, very tall ¤ tall

Note that the building blocks of the above relations (the modifiers, nouns, verbs, etc.) are obtained from the input $P$-$H$ pair. Thus we will not have redundant relations where the words are not in the vocabulary of the pair.

**Relations from input sentences** We further add several patterns that map specific syntactic structures to preorder relations. For example, if the input premise is of the structure: *every noun*$_1$ *is (a/an) noun*$_2$, we add *noun*$_1$ ¤ *noun*$_2$ to $\mathcal{K}$; *John is a man* ÑÏ *all man* ¤ *John* ¤ *man*. Furthermore, *every noun*$_1$ *VP* will be mapped to *be noun*$_1$ ¤ *VP*: *every dog swims* means that if an entity is a dog, then it swims. Such mappings can be extended even further to allow the system to learn more preorder relations from the input premises.

**Exclusion relations** Our $\mathcal{K}$ also includes exclusion relations (Icard, 2012). The antonym and coordinating terms can also be easily extracted from WordNet, as antonyms have been

[4](**At-least** (3)$^{\grave{O}}$)$^{\acute{O}}$ (commissioners)$^{\grave{O}}$ (spend a lot of time at home)$^{\grave{O}}$: $^{\grave{O}}$ on both arguments, same as "some". (**At-most** (10)$^{\grave{O}}$)$^{\acute{O}}$ (commissioners)$^{\acute{O}}$ (spend time at home)$^{\acute{O}}$: $^{\acute{O}}$ on both arguments, same as "no".

| Relation | Examples | Notes |
|---|---|---|
| ⌑ | dog ⌑ animal; dance ⌑ move; small dog ⌑ dog; run fast ⌑ run; all ⌑ some; | Hypernym/hyponym in WordNet |
| K | tall K short, on K off, up K down; | Antonyms in WordNet, ^ in MacCartney's system |
| \| | dog \| cat; sleep \| run; | Coordinating terms in WordNet, \| in MacCartney's system |

Table 4.1: Relations in the knowledge base $\mathcal{K}$

hand-coded, and coordinating terms can be identified as words sharing the same hypernym. These relations are summarized in Table 4.1. Replacing a word with another which is in an exclusion relation with it can result in contradiction: *the dog is running* contradicts *the dog is sleeping*, or a neutral relation: *the man is listening to music* is neutral to *the man is taking a shower*.[5] In the current implementation of `MonaLog`, replacements with an exclusion relation is assumed to result in contradiction only. A more fine-grained classification with respect to exclusion relations may be needed for work on other NLI datasets.

**Implementation details of $\mathcal{K}$** When working on FraCaS and SICK, `MonaLog` does not include all relations from WordNet in the knowledge base $\mathcal{K}$. Rather, it builds a small and customized $\mathcal{K}$ for each NLI problem.

Concretely, for each NLI problem, the premise and hypothesis are first POS-tagged by a POS tagger, and then fed to JIGSAW (Basile et al., 2007) for word sense disambiguation. JIGSAW returns the WordNet id for each word after disambiguation, and this information is used to identify the hypernyms and hyponyms in WordNet. Then, `MonaLog` builds the $\mathcal{K}$ for the premise-hypothesis pair. A `vocab` is constructed from all unique words and phrases in the premise and the hypothesis. The phrases are extracted based on the supertags of the phrases, for instance, *SzNP* would correspond to a VP. Note that for the hypernym/hyponym relations, `MonaLog` adds a relation to $\mathcal{K}$ only if both words are in the

[5] See chapter 5 of MacCartney (2009) for more discussions on this.

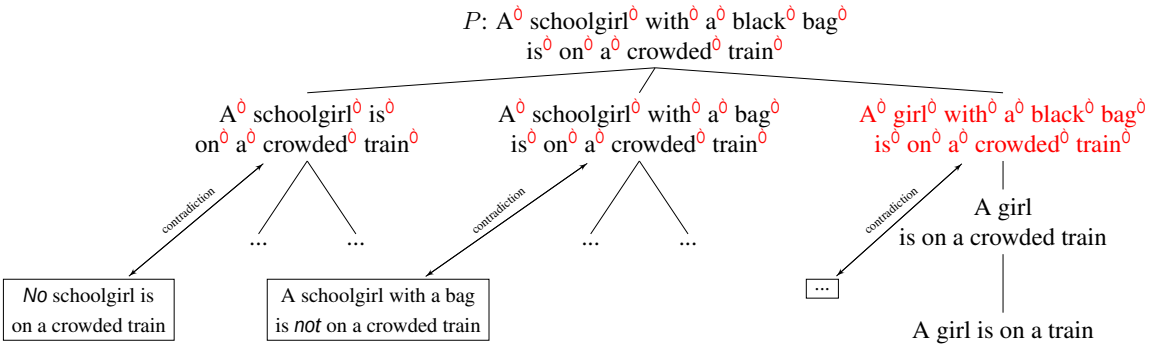



Figure 4.2: Example search tree for SICK 340, where $P$ is *A schoolgirl with a black bag is on a crowded train*, with the $H$: *A girl with a black bag is on a crowded train*. To exclude the influence of morphology, all sentences are represented at the lemma level in `MonaLog`, which is not shown here.

`vocab`. Modifier and exclusion relations are also built based on only the words, phrases in `vocab` and their combinations thereof. Finally, ¤ relations for the quantifiers are manually added to $\mathcal{K}$.

### 4.3.3 Generation

Once we have a polarized CCG tree, and various relations in $\mathcal{K}$, generating entailments and contradictions is fairly straightforward. A concrete example is given in Figure 4.2, which shows the example search tree for SICK 340, where $P$ is *A schoolgirl with a black bag is on a crowded train*, and the $H$ is *A girl with a black bag is on a crowded train*. In our implementation, only one `replacement` is allowed at each step. Sentences at the nodes in the search tree in the Figure are the generated entailments. Sentences in rectangles are the generated contradictions. In the example of Figure 4.2, our system will return `entail`. The search will terminate after reaching the $H$ in this case, but for illustrative purposes, we show entailments of depth up to 3.

Note that the generated ¤ instances are capable of producing mostly monotonicity inferences, but `MonaLog` can be extended to include other more complex inferences in *natural logic*, hence the name Mo*na*Log. This extension is described in chapter 4.4.

**Entailed sentences** The key operation for generating entailments is `replacement`, or substitution. It can be summarized as follows: 1) For upward-entailing (UE) words or constituents, replace them with words or constituents that denote larger sets. 2) For downward-entailing (DE) words/constituents, either replace them with those denoting smaller sets, or add modifiers (adjectives, adverbs and/or relative clauses) to create a smaller set. Thus for *every*$^{\grave{0}}$ *linguist*$^{\acute{0}}$ *swim*$^{\grave{0}}$, `MonaLog` can produce the following three entailments by replacing each word with the appropriate word from $\mathcal{K}$: *most linguist swim*, *every semanticist swim* and *every linguist move*. These are results of one `replacement`. Performing `replacement` for multiple rounds/depths can easily produce many more entailments.

**Contradictory sentences** To generate sentences contradictory to the input sentence, we do the following: 1) if the sentence starts with "no (some)", replace the first word with "some (no)". 2) If the object is quantified by "a/some/the/every", change the quantifier to "no", and vice versa. 3) Negate the main verb or remove the negation. See examples in Figure 4.2.

**Neutral sentences** `MonaLog` returns Neutral if it cannot find the hypothesis $H$ in the generated entailments or contradictions ($\mathcal{S}.entailments$ or $\mathcal{S}.contradictions$). Thus, there is no need to generate neutral sentences. However, it is possible to generate neutral statements with `MonaLog`. In essence, a neutral statement can be generated by performing replacement in the opposite direction of the monotonicity arrows. In other words, we can: 1) Replace a DE word/constituent by another denoting a bigger set: *every linguist*$^{\acute{0}}$ *swim* is neutral to *every person swim*, because *linguist* ¤ *person*. 2) Replace a UE token/constituent by another one denoting a smaller set, which is the opposite of 1): *every linguist move*$^{\grave{0}}$ is neutral to *every linguist swim* because *swim* ¤ *move*.

It is worth noting is that "neutral" can be understood as the default category where any pairs that are not entailment or contradiction will be classified as. Thus depending on the construction of the dataset, neutral pairs may include many different cases. For example,

in SICK, reverse-entailment (i.e., $H$ entails $P$) as well as unrelated pairs will be labelled as "neutral". The neutral pairs that can be generated by `MonaLog` are in essence reverse-entailments, and it is not possible for `MonaLog` to generate unrelated pairs.

### 4.3.4 Search

Now that we have a set of inferences and contradictions stored in $\mathcal{S}$, we can simply see if the hypothesis is in either one of the sets by comparing the strings. If yes, then return Entailment or Contradiction; if not, return Neutral, as schematically shown in Figure 4.2. However, the exact-string-match method is too brittle. For instance, "man on the street", "a man on the street" and "the man on the street" would be considered equivalent in most NLI datasets (Samuel R Bowman et al., 2015; M. Marelli et al., 2014). Therefore, we apply a heuristic. If the only difference between sentences $S_1$ and $S_2$ is in the set {"a", "be", "ing"}, then $S_1$ and $S_2$ are considered semantically equivalent.

The search is implemented using depth first search, with a default depth of 2, i.e. at most 2 replacements for each input sentence. At each node, `MonaLog` "expands" the sentence (i.e., an entailment of its parent) by obtaining its entailments and contradictions, and checks whether $H$ is in either set. If so, the search is terminated; otherwise the systems keeps searching until all the possible entailments and contradictions up to depth 2 have been visited.

Next, we will report two experiments using `MonaLog`, one on the FraCaS dataset (Cooper et al., 1996), and the other on the SICK dataset (M. Marelli et al., 2014).

## 4.4 Experiment on FraCaS

### 4.4.1 Experimental Setting

In FraCaS, only section one focuses on monotonicity, which is the section we test `MonaLog` on.[6] There are a total of 74 NLI problems in section one, 30 of which have more than one premise (see examples in Table 2.3). In this subsection, we will first discuss how `MonaLog` handles multi-premise problems, and then how natural logic rules beyond monotonicity are incorporated in `MonaLog`.

First, to deal with problems with multiple premises, `MonaLog` simply loops through each premise and generates entailments, neutrals and contradictions based on that premise. All generated statements will be added to the sentence base $\mathcal{S}$ for the final decision making. We also update the knowledge base $\mathcal{K}$ when reading through the premises. For instance, the second premise of FraCaS-026, *all Europeans are people*, will result in a new preorder *Europeans* ⌑ *people*, which will be added to $\mathcal{K}$.

Second, as monotonicity inference covers only a subset of all natural logic phenomena, in order to work on problems involving other natural logic inferences, `MonaLog` needs to be augmented with other natural logic rules. Below we give two example natural logic rules which cannot be handled by monotonicity and the replacement operation alone and describe how they are incorporated into the system.

$$\frac{\textit{Some y are x}}{\textit{Some x are y}}\ \text{SOME}_2 \qquad \frac{\textit{Det x are y} \qquad \textit{All x are z}}{\textit{Det x are (y ^ z)}}\ \text{DET}$$

$\text{SOME}_2$ states that if *Some y are x*, then *Some x are y*. This is straightforward in terms of set relations; both sentences state that the intersection of the set of $x$ and the set of $y$ is non-empty.

In the DET rule, the shared *Det* in the first statement of the premise and the conclusion can be *every*, *some* and *most*. We sketch an informal proof when *Det* is *some* below, and

[6] This subsection is based on H. Hu et al. (2019). Other sections of FraCaS include: anaphora, plurals, comparatives, etc., which are beyond the design and capabilities of `MonaLog`.

the proof for other determiners can be obtained in a similar manner. Our assumptions are: *some x are y*, *all x are z*. Our goal is to prove *some x are both y and z*. From the first assumption (*some x are y*), we know that there must be some $n$ which satisfies $n$ P $x$ as well as $n$ P $y$. From the second assumption (*all x are z*) and $n$ P $x$, we infer that $n$ P $z$. Thus we have $n$ P $y$ and $n$ P $z$, which is equivalent to $n$ P p$y$ ^ $z$q and also what we aim to prove. The DET rule is needed to solve FraCaS-026, which we will explain in the next section.

On the implementation side, to incorporate the natural logic rules, we first represent simple premises in a specific format, which is compatible to natural logic syntax of the form *quantifier x y*. For instance, *Most Europeans are resident in Europe* is represented as *most (Europeans) (resident in Europe)*. Because natural logic rules such as $\text{SOME}_2$ and DET are also represented in this form, we can then make inferences based on the rules. Finally we convert sentences in natural logic to sentence in natural language. This usually only involves minimal editing. For example, *every* cat *(animal ^ meow)* will be converted to "every cat is an animal who meows". As we will show in the next section, DET is useful in solving several multi-premise problems in the first section of FraCaS. Note that the capacity of our system can be further expanded by incorporating more rules from natural logic.

### 4.4.2 Results and Discussion

This section reports the performance of `MonaLog` on the first section of FraCaS: generalized quantifiers.[7] The results are shown in Table 4.2 and Table 4.3. We have perfect precision for Entailment and Contradiction (see Table 4.3), which means the entailed and contradictory hypotheses detected by our systems are all correct. Table 4.2 also shows comparable accuracy with previous systems (ours: 88%, previous systems: 62–95%).[8] Our errors come from cases where the system fails to classify a pair into either entailment or

[7] To reproduce the results, refer to: https://github.com/huhailinguist/monalog/blob/master/src/fracas.py.
[8] All the previous systems in Table 4.2 are reviewed in chapter 2.2.

| system | MM08 | AM14 | LS13 | T14 | D14 | M15 | A16 | ours |
|---|---|---|---|---|---|---|---|---|
| multi-premise? | N | N | Y | Y | Y | Y | Y | Y |
| # problems | 44 | 44 | 74 | 74 | 74 | 74 | 74 | 74 |
| Acc. (%) | 97.73 | 95 | 62 | 80 | 95 | 77 | 93 | 88 |

Table 4.2: Accuracy of our system and previous ones. MM08: MacCartney and Christopher D Manning (2008). AM14: Angeli and C. Manning (2014). LS13: M. Lewis and Steedman (2013). T14: Tian et al. (2014). D14: Dong et al. (2014). M15: Mineshima et al. (2015). A16: Abzianidze (2016b).

| Truth / Pred | E | U | C |
|---|---|---|---|
| Entail | 29 | 7 | 0 |
| Unknown | 0 | 33 | 0 |
| Contradict | 0 | 2 | 3 |

Table 4.3: Confusion matrix of our system. Our system achieves 100% precision.

contradiction. Several of these errors have to do with difficult syntactic structures for a rule-based system, such as "one of + noun phrase". For instance, in fracas-014,

Premise 1: *Neither leading tenor comes cheap.*

Premise 2: *One of the leading tenors is Pavarotti.*

Hypothesis: *Pavarotti is a leading tenor who comes cheap.*

`MonaLog` fails to make the correct prediction of "contradiction" because it has no means to deal with "one of". Theoretically, one can write custom rules that translate input sentences with different syntactic structures into the pre-order representation in our $\mathcal{K}$. For this example, the following rule would suffice: "one of the $\text{NOUN}_1$ is $\text{NOUN}_2$" Ñ $\text{NOUN}_2$ ¤ $\text{NOUN}_1$. However, it is not feasible or practical to specify a rule for every possible syntactic variation. This highlights the challenge for a rule-based system to have a high recall given the vast and almost unlimited syntactic variations of natural language. We discuss these limitations in chapter 4.6.

**An example** The following demonstrates the steps for solving FraCas-026, which is a multiple-premise problem. To solve this problem, we need both the replacement operation and also the DET rule.

1. Obtain monotonicity arrows/polarities[9] of all premises $\mathcal{P}$, but not the hypothesis $H$:

   Premise 1: Most$^{\grave{0}}$ Europeans are$^{\grave{0}}$ resident$^{\grave{0}}$ in$^{\grave{0}}$ Europe$^{\grave{0}}$

   Premise 2: All$^{\grave{0}}$ Europeans$^{\acute{0}}$ are$^{\grave{0}}$ people$^{\grave{0}}$

   Premise 3: All$^{\grave{0}}$ people$^{\acute{0}}$ who$^{\acute{0}}$ are$^{\acute{0}}$ resident$^{\acute{0}}$ in$^{\acute{0}}$ Europe$^{\acute{0}}$ can$^{\grave{0}}$ travel$^{\grave{0}}$ freely$^{\grave{0}}$ within$^{\grave{0}}$ Europe$^{\acute{0}}$

   Hypothesis: Most Europeans can travel freely within Europe

2. Update knowledge base $\mathcal{K}$ with the information from $\mathcal{P}$. For example, based on Premise 3, which has the form "every (or equivalent quantifiers, see chapter 4.3.2) X VP", we add to the knowledge base:

   $$\textit{be people who be resident in Europe} \sqsupset \textit{can travel freely within Europe} \tag{4.5}$$

   Note that all words are represented as their lemmas in order for the system to be less brittle and account for morphological variations.

3. Using the DET rule, the system generates a series of sentences which will also be polarized. For instance, applying the DET rule to P1 and P2:

$$\frac{\textit{P1: Most Europeans are resident in Europe} \qquad \text{P2: } \textit{All Europeans are people}}{\textit{Most Europeans are (resident in Europe ^ people)}}\ \text{DET}$$

[9] The polarity marking in P3 of the second occurrence of *Europe* was corrected from our system's output. The point is that under the scope of a modal *can*, a prepositional phrase headed by *in* or *within* changes polarity. To see this, Premise 3 entails that "All people who are resident in Europe can travel freely within *France*", and it does not entail they can travel freely within *Europe and Asia*.

we obtain

*Most*$^{\Diamond}$ *Europeans* *be*$^{\Diamond}$ *people*$^{\Diamond}$ *who*$^{\Diamond}$ *be*$^{\Diamond}$ *resident*$^{\Diamond}$ *in*$^{\Diamond}$ *Europe*$^{\Diamond}$ (4.6)

*Most*$^{\Diamond}$ *Europeans* *be*$^{\Diamond}$ *resident*$^{\Diamond}$ *in*$^{\Diamond}$ *Europe*$^{\Diamond}$ *who*$^{\Diamond}$ *be*$^{\Diamond}$ *people*$^{\Diamond}$ (4.7)

4. Then `MonaLog` adds generated sentences into sentence base $\mathcal{S}$.entailments, and starts to perform `replacement` on every constituent of each sentence. We obtain a list of entailed statements:

   *Many European be people who be resident in Europe* (most ¤ many)

   *Every European be people* (all = every)

   *Most European can travel freely within Europe* (from (4.5) and (4.6))

   *Some European be people who be resident in Europe* (most ¤ some)

   ...

   We also generate contradictions:

   *No European be people who be resident in Europe* (per rules for contra. generation)

   *No European be people* (per rules for contra. generation)

   ...

5. Lastly, one of the sentences in the list of inferences above (the one underlined) matches the given hypothesis *H*, which means that `MonaLog` will return `entail`. Note that the generation of entailments and contradictions will halt if one of the generated sentences match the hypothesis.

**Choice of parsers and their errors** Parser performance is a major bottleneck of the system. We have tested two commonly used CCG parsers, C&C (S. Clark and Curran,

| id | premise | hypothesis | orig. label | corr. label |
|---|---|---|---|---|
| 219 | There is no girl in white dancing | A girl in white is dancing | C | C |
| 294 | Two girls are lying on the ground | Two girls are sitting on the ground | N | C |
| 743 | A couple who have just got married are walking down the isle | The bride and the groom are leaving after the wedding | E | N |
| 1645 | A girl is on a jumping car | One girl is jumping on the car | E | N |
| 1981 | A truck is quickly going down a hill | A truck is quickly going up a hill | N | C |
| 8399 | A man is playing guitar next to a drummer | A guitar is being played by a man next to a drummer | E | n.a. |

Table 4.4: Examples from SICK (M. Marelli et al., 2014) and corrected SICK (Kalouli et al., 2017b, 2018) w/ syntactic variations. n.a.: example not checked by Kalouli and her colleagues. C: contradiction; E: entailment; N: neutral.

2007) and EasyCCG (M. Lewis and Steedman, 2014). C&C fails to parse four sentences from section one of FraCaS. EasyCCG can parse all of them but we still need to semi-automatically modify the trees. Some of these are modifications that transform the tree into a semantically more meaningful form, while others are correcting parse errors. For example, not all quantifiers are super-tagged consistently: *most* and *few* are sometimes tagged as `N/N` when they should be tagged as `NP/N` like other quantifiers. There are also parsing errors involving multi-word expressions such as "a lot of", "used to be". At the moment, `MonaLog` can only correct the systematic errors from the parsers.

## 4.5 Experiment on SICK

In this subsection, we report results of `MonaLog` on SICK.[10] As a reminder, we list again some examples from the SICK corpus in Table 4.4.

### 4.5.1 Experimental Setting

**Setup** The goal of this experiment is to test how accurately `MonaLog` solves problems in a large-scale dataset. We first used the system to solve the 495 problems in the trial set

[10]This is based on section 4 of H. Hu et al. (2020a).

and then manually identified the cases in which the system failed. Then we determined which syntactic transformations are needed for MonaLog. After improving the results on the trial data by introducing a preprocessing step to handle limited syntactic variation (see below), we applied MonaLog on the test set. This means that the rule base of the system was optimized on the trial data, and we can test its generalization capability on the test data.

**Pre-processing** As discussed before, one main obstacle for MonaLog is the syntactic variations in language data, illustrated in some examples of SICK in Table 4.4. There exist multiple ways of dealing with these variations: One approach is to 'normalize' unknown syntactic structures to a known structure. For example, we transform passive sentences into active ones and convert existential sentences into the base form (see ex. 8399 and 219 in Table 4.4). Another approach is to use some more abstract syntactic/semantic representation so that the linear word order can largely be ignored, for example, represent a sentence by its dependency parse, or use Abstract Meaning Representation (AMR). We opted for the first option because our polarized sentences returned by ccg2mono are represented in CCG trees; thus in order to utilize the polarities of the sentences we cannot adopt another representation such as the dependency parse or AMR. We believe that dealing with a wide range of syntactic variations requires tools designed specifically for that purpose. The goal of MonaLog is to generate entailments and contradictions based on a polarized sentence instead.

Below, we list the most important syntactic transformations we perform in preprocessing using Python scripts.

1. Convert all passive sentences to active using *pass2act*[11]. If the passive does not contain a *by* phrase, we add *by a person*.
2. Convert existential clauses into their base form (see ex. 219 in Table 4.4).
3. Convert simple relative clause to an attribute, for instance, *A man in **a jersey which***

[11] https://github.com/DanManN/pass2act

***is black*** *is standing in a gym* Ñ *A man in* ***a black jersey*** *is standing in a gym* (SICK 9132).

4. Multi-word quantifiers to a single-word counterpart for better parse trees from the CCG parsers: *a few* Ñ *several*, *a group of* Ñ *some*, *a lot of* Ñ *much*.
5. Other transformations: *someone/anyone/no one* Ñ *some/any/no person*; *there is no man doing sth.* Ñ *no man is doing sth.*; *blue colored jacket* Ñ *blue jacket*.

**Corrected SICK** As mentioned in chapter 2.1.3, there are numerous issues with the original SICK dataset. In particular, some of the inference labels are wrongly annotated, as illustrated by Kalouli et al. (2017b, 2018). They manually went through roughly 3,000 out of the 10,000 examples in SICK, and noted that 409 pairs have been wrongly annotated. For this reason, my advisor Dr. Moss and I checked the 409 changes. We found that only 246 problems are labeled the same by our team and by Kalouli et al. (2018). For the 163 cases where there is disagreement, we adjudicated the differences after a discussion.

We are aware that the partially checked SICK (by two teams) is far from ideal. We therefore present results for two versions of SICK: the original SICK and the version corrected by our team.

### 4.5.2 Results on of `MonaLog` on SICK

The results of our system on uncorrected and corrected SICK are presented in Table 4.5 and Table 4.6 respectively, along with comparisons with other systems.

Our accuracy on the uncorrected SICK (77.19%) in Table 4.5 is much higher than the majority baseline (56.36%) or the hypothesis-only baseline (56.87%) reported by Poliak et al. (2018), and 4 to 7 points lower than current logic-based systems. Since our system is based on *natural logic*, there is no need for translation into logical forms, which makes the reasoning steps transparent and much easier to interpret. I.e., with entailments and contradictions, we can generate a natural language trace of the system, see Fig. 4.2.

| system | P | R | acc. |
|---|---|---|---|
| majority baseline | – | – | 56.36 |
| hypothesis-only baseline | – | – | 56.87 |
| Poliak et al. (2018) | | | |
| MonaLog (this work) | | | |
| MonaLog + all transformations | 83.75 | 70.66 | 77.19 |
| Hybrid: MonaLog + BERT | 83.09 | 85.46 | 85.38 |
| ML/DL-based systems | | | |
| BERT (base, uncased) | 86.81 | 85.37 | 86.74 |
| Yin and Schütze (2017) | – | – | **87.1** |
| Beltagy et al. (2016) | – | – | 85.1 |
| Logic-based systems | | | |
| Bjerva et al. (2014) | 93.6 | 60.6 | 81.6 |
| Abzianidze (2015) | 97.95 | 58.11 | 81.35 |
| Martínez-Gómez et al. (2017) | 97.04 | 63.64 | 83.13 |
| Yanaka et al. (2018) | 84.2 | 77.3 | 84.3 |

Table 4.5: Performance on the **uncorrected** SICK test set. P / R for MonaLog averaged across three labels. Results involving BERT are averaged across six runs; same for later experiments.

Our results on the corrected SICK are presented in Table 4.6. MonaLog's accuracy is 4 points higher than on the uncorrected SICK, reaching 81.66%, demonstrating the effect of data quality on the final results. Note that with some simple syntactic transformations mentioned in the previous section we can gain 1-2 points in accuracy (see Table 4.6).

Table 4.7 shows MonaLog's performance on the individual relations. The system is clearly very good at identifying entailments and contradictions, as demonstrated by the high precision values, especially on the corrected SICK set (98.50 precision for E and 95.02 precision for C). The lower recall values are due to MonaLog's current inability to handle syntactic variation.

### 4.5.3 Hybridizing MonaLog with BERT

Based on the results above, we tested a hybrid model of MonaLog and BERT (see Table 4.5) where we exploit MonaLog's strength: Since MonaLog has a very high precision

| system | P | R | acc. |
|---|---|---|---|
| MonaLog + existential trans. | 89.43 | 71.53 | 79.11 |
| MonaLog + pass2act | 89.42 | 72.18 | 80.25 |
| MonaLog + all transformations | 89.91 | 74.23 | 81.66 |
| Hybrid: MonaLog + BERT | 85.65 | 87.33 | **85.95** |
| BERT (base, uncased) | 84.62 | 84.27 | 85.00 |

Table 4.6: Performance on the **corrected** SICK test set.

| | Entailment | | Contradiction | | Neutral | |
|---|---|---|---|---|---|---|
| | P | R | P | R | P | R |
| uncorr. SICK | 97.75 | 46.74 | 80.06 | 70.24 | 73.43 | 94.99 |
| corr. SICK | 98.50 | 50.46 | 95.02 | 73.60 | 76.22 | 98.63 |

Table 4.7: Results of MonaLog per label.

on Entailment and Contradiction, we can always trust MonaLog if it predicts E or C; when it returns N, we then fall back to BERT. This is a proof-of-concept experiment to see if the performance on the neutral examples can be improved, assuming that there are many neutral pairs that MonaLog is not equipped to handle, since it can only generate entailments and contradictions. This hybrid model improves the accuracy of BERT by 1% absolute to 85.95% on the corrected SICK. On the uncorrected SICK dataset, the hybrid system performs worse than BERT by about 1.4%. Since MonaLog is optimized for the corrected SICK, it may mislabel many E and C judgments in the *uncorrected* dataset. The standalone BERT system performs better on the uncorrected data (86.74%) than the corrected set (85.00%). The corrected set may be too inconsistent since only a third of the full dataset has been checked.

Overall, these hybrid results show that it is possible to combine our high-precision system with deep learning architectures. However, more work is necessary to optimize this combined system.

| id | premise | hypothesis | SICK | corr. SICK | Mona |
|---|---|---|---|---|---|
| 359 | There is no dog chasing another or holding a stick in its mouth | Two dogs are running and carrying an object in their mouths | N | n.a. | C |
| 1402 | A man is crying | A man is screaming | N | n.a. | E |
| 1760 | A flute is being played by a girl | There is no woman playing a flute | N | n.a. | C |
| 2897 | The man is lifting weights | The man is lowering barbells | N | n.a. | C |
| 2922 | A herd of caribous is not crossing a road | A herd of deer is crossing a street | N | n.a. | C |
| 3403 | A man is folding a tortilla | A man is unfolding a tortilla | N | n.a. | C |
| 4333 | A woman is picking a can | A woman is taking a can | E | N | E |
| 5138 | A man is doing a card trick | A man is doing a magic trick | N | n.a. | E |
| 5793 | A man is cutting a fish | A woman is slicing a fish | N | n.a. | C |

Table 4.8: Examples of incorrect answers by MonaLog; n.a. = the problem has not been checked in corr. SICK.

### 4.5.4 Error Analysis

Upon closer inspection, some of MonaLog's errors consist of difficult cases, as shown in Table 4.8. For example, in ex. 359, if our knowledge base $\mathcal{K}$ contains the background fact *chasing* ⌑ *running*, then MonaLog's judgment of C would be correct. In ex. 1402, if *crying* means *screaming*, then the label should be E; however, if *crying* here means *shedding tears*, then the label should probably be N. Here we also see potentially problematic labels (ex. 1760, 3403) in the original SICK dataset.

Another point of interest is that 19 of MonaLog's mistakes are related to the antonym pair *man* vs. *woman* (for example, ex. 5793 in Table 4.8). This points to inconsistency of the SICK dataset: Whereas there are at least 19 cases tagged as Neutral (for example, ex. 5793), there are at least 17 such pairs that are annotated as Contradictions in the test set (for example, ex. 3521), P: *A man is dancing*, H: *A woman is dancing* (ex. 9214), P: *A shirtless man is jumping over a log*, H: *A shirtless woman is jumping over a log*. If *man* and *woman* refer to the same entity, then clearly that entity cannot be *man* and *woman* at the

same time, which makes the sentence pair a contradiction. If, however, they do not refer to the same entity, then they should be Neutral.[12] This highlights the importance of having a consistently annotated corpus, with clear instructions on co-reference. We will have more discussions on the annotation issues next.

## 4.6 Discussion and Limitations

In this subsection, we discuss some of the challenges and issues in MonaLog and NLI, and point to a few issues that would benefit from more investigation.

### 4.6.1 Syntactic variation

As discussed in Chapter 4.5.1, MonaLog has to rely on a set of pre-defined syntactic transformation rules to handle the unlimited syntactic variations in human language. However, this is unlikely to scale up to larger, crowd-sourced data such as SNLI/MNLI. This is a fundamental issue for logic-based models. That is, how to map different ways of expressing the same meaning into a unified meaning representation. A potential future direction is to use distributional models such as word embeddings to represent the meaning and monotonicity and in essence have a neural-based MonaLog. Designing a neural-based system that is capable of monotonicity reasoning and can deal with the boundless variation in syntax will be a major contribution to the field. There are some developments in this line of work recently (E. Chen et al., 2021).

### 4.6.2 Issues in NLI Annotation

Annotation for NLI pairs has been shown to be a non-trivial issue (Geva et al., 2019; Nie et al., 2020b; Pavlick and Kwiatkowski, 2019), where the disagreement among annotators is shown to be intrinsic to the task itself, rather than an artifact that can be easily remedied,

[12] It is not clear from the SICK paper whether they told the annotators that the sentences are descriptions of images (in SNLI, this is explicitly mentioned in the instructions). If it was the case, then the pairs that only differ in one word (man vs. woman) should be annotated as contradiction.

for example, by having (much) more annotators labelling the same pair or providing longer context to the sentence pair.

Here we will mention some issues we encountered in an ongoing work that aims to have a fully checked SICK dataset.

**Co-reference** When should annotators make the entities in the premise and hypothesis co-refer? This has been discussed at length, at least as early as in de Marneffe et al. (2008) where they argued that "compatible noun phrases between sentences are assumed to be coreferent in the absence of clear countervailing evidence." But whether two noun phrases are "compatible" seems to be a judgmental call.

For example, SICK 522:

$P$ : *A woman who is wearing a pink boa is riding a bicycle down a bridge built for pedestrians.*

$H$ : *The woman wearing silver pants, pink bellbottoms and a pink scarf isn't riding a bike.*

The question is whether the woman can be wearing a pink boa (from the premise) and silver pants + pink bellbottoms + pink scarf (from the hypothesis) at the same time. If yes, then "the woman" in H and P could be said to co-refer (they are "compatible"), and we would have a contradiction. However, if the premise and the hypothesis are talking about two different women, then we should have a neutral and also unrelated pair.

In fact, de Marneffe et al. (2008) and SNLI (Samuel R Bowman et al., 2015) go one step further to emphasize "event coreference", and attempt to maximally make the premise and hypothesis refer to the same event. Therefore, in SNLI, it is assumed that P and H describe the same event, and it is thus argued that *Ruth Bader Ginsburg was appointed to the US Supreme Court* and *I had a sandwich for lunch today* should be labelled as contradiction

because the Supreme Court appointment and a person having a sandwich cannot be the same event. However, it is against most people's intuition that such *unrelated* pairs should be labelled as neutral.

At the end of the day, assuming some degree of entity coreference seems to be a requirement in NLI annotation, but whether to strictly enforce event coreference is a more subtle issue. One solution is to filter out the "unrelated" pairs in advance, which is done in de Marneffe et al. (2008), and partially addressed in SICK where each pair is also scored for their relatedness. However, the entailment label and the relatedness score have been unfortunately treated in a disconnected fashion in most studies involving SICK. The Original Chinese NLI (OCNLI) dataset in Chapter 6 takes a similar approach to SICK and provides a fourth label (for "unrelated" pairs) to the annotators, but these labels turn out to be extremely rare. How entity/event coreference figures into inference and relatedness in the NLI task needs to be further investigated, potentially by asking the annotators explicitly whether they think the entities/events corefer.

**Difficulty in lexical semantics** Interestingly, during the check of SICK examples, there have been many disagreements on lexical semantics. (The same is reported in Pavlick and Kwiatkowski (2019).) For example, is cutting the same as slicing? Is a kettle the same as a pot? Is a pan the same as a pot? Is a girl a woman and vice versa? What counts as idling? Are playing kids idling? Is walking on the street a kind of idling? These are questions that annotators are unlikely to completely agree on. There is still no consensus of how to deal with these cases in NLI.

As mentioned in Pavlick and Kwiatkowski (2019), one can sometimes observe a bimodal distribution over the inference labels. In one of their examples, some annotators believe that *Paula swatted the fly* entails that *The swatting happened in the forceful manner* (entailment); however, many among the same group of annotators rate the example as neutral, suggesting that they may believe that the swatting didn't necessarily happen in

a forceful manner. This example suggests that by simply having more annotators may not always result in a more convergent annotation.

Some have proposed that instead of treating NLI as a 2-way or 3-way classification task, we can ask the annotators to estimate a confidence score that indicates how likely the premise entails the hypothesis, which we will briefly introduce now.

**Another view of NLI** There are attempts to frame the NLI task in a different way. For instance, T. Chen et al. (2020) asks the annotators to give a *confidence score* about whether the premise entails the hypothesis, instead of a 3-way classification label, making NLI a regression task rather than a classification one. Pavlick and Kwiatkowski (2019) and Nie et al. (2020b) trained models to model the distribution of human-annotated labels. How to incorporate the subtleties of human judgments into the NLI task and devise better evaluation method is an important area for future research.

### 4.6.3 What kind of NLI datasets should we evaluate on?

Should logic-based systems be evaluated on a logic-oriented dataset, rather than crowd-sourced SNLI/MNLI? Is it unfair to compare logic-based (or symbolic) systems with deep learning models as the current large-scale datasets are inherently favoring the deep learning models? This also has to do with our expectation of the logic-based models. If they are designed to solve logical inferences, rather than more pragmatic/world-knowledge oriented inferences, then we should design more datasets such as SICK or FraCaS and focus on specific logical phenomena, instead of hoping for the logic-based models to perform well on datasets such as SNLI and MNLI.

## 4.7 Summary

In this chapter, we described `MonaLog`, a program that can make inferences based on monotonicity and natural logic. we reported empirical results on two NLI/RTE datasets,

FraCaS and SICK. The results show that `MonaLog` achieves comparable performance to other logic-based models, despite its light-weight design, which does not need translation from input text to logical forms. we also discussed issues in NLI annotation and pointed out areas that need more investigation.

# CHAPTER 5

# A HIGH-QUALITY DATASET FOR CHINESE NATURAL LANGUAGE INFERENCE

In this chapter and the next one, we shift our focus from symbolic models to neural models, and our language of attention from English to Chinese. This is driven by at least the following reasons. First, in the past decade, it has become increasingly clear that having a good symbolic model is not enough to deal with the flexible and diverse natural language "in the wild", which is a key feature in NLI datasets leaning towards the more informal definition of NLI (Samuel R Bowman et al., 2015; Williams et al., 2018). Second, neural models such as BERT (Devlin et al., 2019) pre-trained with large quantities of text and then fine-tuned on human-labeled data have shown tremendous success in many natural language understanding tasks in English, including NLI. Multilingual versions of these neural models have sometimes been argued or shown to be able to solve NLU problems in many languages at the same time with one single model (Conneau et al., 2020; Goyal et al., 2021; Xue et al., 2020).

However, for NLP researchers in Chinese, there are no high-quality training or evaluation data for NLI, making it impossible to examine whether these neural models will work in one of the most widely spoken languages in the world. Furthermore, the lack of resources also means that we cannot test whether the multilingual neural models can perform well on two different languages (English and Chinese) at the same time. These are important questions for the neural models, since it has been assumed that one of their advantages is that their representation of language units—vectors rather than strings—is language-independent, and should in theory be applicable to all human languages.

Under this background, this chapter introduces the first high-quality, large-scale NLI dataset for Chinese, collected using an enhanced annotation procedure guided by research

on the quality and problems of previously created datasets in English.[1] We then experiment with a variety of neural models to see how they perform on our newly collected corpus. We also discuss the effects of different annotation procedures for data quality. The next chapter will report a study on the ability multilingual models for Chinese NLI.

**Structure of this chapter:** 1) After a brief review of the issues in previous NLI datasets and motivations for our Chinese dataset (section 5.1), and laying out the research questions (section 5.2), a new, high quality dataset for NLI for Chinese will be introduced, based on Chinese data sources and expert annotators (section 5.3); 2) We provide strong baseline models for the task, and establish the difficulty of our task through experiments with recent pre-trained transformers (section 5.4). 3) We also demonstrate the benefit of naturally annotated NLI data by comparing performance with large-scale automatically translated datasets (section 5.4). 4) We present an analysis of different subsets and genres of OCNLI (section 5.5 and section 5.6).

## 5.1 Issues in Previous NLI Datasets and Motivation for OCNLI

In this section, I briefly review the issues that have been found in existing NLI datasets and motivate the creation of OCNLI. For a more complete review of NLI resources, refer to chapter 2.1.

### 5.1.1 Previous NLI Datasets in English

The datasets that are most relevant for the work in this chapter are the two large-scale, human-elicited datasets in English: the Stanford Natural Language Inference Corpus (SNLI) (Samuel R Bowman et al., 2015), whose premises are taken from image captions, and the Multi-Genre Natural Language Inference Corpus (MNLI) (Williams et al., 2018), whose

[1] Chapters 5.1 – 5.4 are based on H. Hu et al. (2020b). I designed the data collection procedure and collected all the data. The experiments in chapter 5.4 are run together with Kyle Richardson. Experiments in chapter 5.5 and 5.6 are run by myself.

premises are from texts in 10 different genres. Both are built by collecting premises from pre-defined text, then having annotators write the hypotheses and give inference labels, which is the procedure we employ and improve upon in our work.

As mentioned in chapter 2.1.4, there are two main issues in SNLI and MNLI.

First, the datasets are not reflective of the true reasoning abilities of NLU models. That is, models that achieve high performance on these datasets still fail catastrophically in real-world applications (Ribeiro et al., 2020), and NLP practitioners generally agree that our current models are still far from achieving human-like understanding ability despite the "super-human" performance on various tasks (Kiela et al., 2021). On the one hand, we expect the evaluation data in NLP to reflect the models' reasoning ability in real-world applications; thus scoring high on the evaluation data but exhibiting low reasoning ability means that the evaluation data are not representative of the challenges in real-world natural language data. On the other hand, we also expect our training data to be able to train robust models. This of course needs to be measured through experimentation, but intuitively, a training set including examples of a diverse range of linguistic phenomena should be helpful.

Then the question in this chapter is how to make our Chinese corpus more representative of the real challenges of natural language data, and at the same time collect a more diverse set of examples. We explain in detail our enhanced procedure of the MNLI-style data collection in chapter 5.3.2.

The second issue is that SNLI and MNLI contain biases that the models can exploit to achieve high scores without really learning to perform NLI. Clearly, in an ideal case, we expect our datasets to contain no biases and models trained on them are learning to reason rather than attending to the superficial/spurious clues that are not representative of regularities in natural language data to achieve high scores. For this issue, the question is whether our enhanced procedure can reduce the aforementioned biases. We closely monitor the hypothesis-only bias and conduct experiments to quantify it in chapter 5.5.2. We also

update our instructions in the annotation process, as described in chapter 5.3.2 to control for lexical overlap. While the biases are a main concern for NLI data collection, our first-and-foremost goal is to ensure that our dataset is representative of the real world inferences for the neural models to learn from, as this is only our first step towards building a Chinese NLI dataset from scratch.

### 5.1.2 Previous NLI Datasets in Other Languages

As reviewed in chapter 2.1, many of the non-English NLI resources are translations from data in English, which may not be ideal for reasons of translation quality and cultural incompatibility. We already mentioned in chapter 2.1 the poor translation quality of the Chinese portion of XNLI (Conneau et al., 2018b), as well as the lack of data reflecting the Chinese culture and context. These concerns motivate us to create a Chinese NLI dataset without using any translations at all.

Furthermore, in this chapter, we also perform a quantitative study to compare the quality of the Chinese portion of XNLI and a randomly chosen part of our OCNLI corpus. We defer the details of the comparison to chapter 5.3.4, after we have introduced the annotation process of OCNLI. At this point, we will give a summary of our findings: Annotators who are native speakers of Chinese were asked to label the inference labels for 400 pairs of mixed XNLI examples and examples from our collected data. The agreement between the annotator label and the gold label is 67% for the XNLI examples, but 84% for our examples. The XNLI examples also have a much higher chance of being labelled as "unrelated" (125 vs. 22) or "incomprehensible" (22 vs. 40). All these results suggest that OCNLI is a more reliable resource for Chinese NLI.

## 5.2 Research Questions

In this chapter we ask the following research questions.

1. How can we improve the annotation methods used in SNLI/MNLI to create more

challenging datasets? Is there a way to encourage or force the annotator to be more creative during annotation?

2. Is the resulting corpus OCNLI challenging to current transformer models? Are there any differences among the data collected using 4 different annotation procedures and instructions? Since our texts are extracted from sources in 5 different genres, are there any differences in how challenging the NLI pairs are among the 5 genres? We explain in greater detail the 4 subsets in chapter 5.3.2 and 5 genres in chapter 5.3.1.

3. Is OCNLI of higher-quality than the Chinese portion of XNLI? We compare annotators agreement on examples randomly sampled from both datasets.

4. Are the data collected using our enhanced methods better training data for the neural models, evaluated on a out-of-domain dataset?

OCNLI has been a major contribution in NLU resource for Chinese (Xu et al., 2020), and has implications on how to improve the annotation and crowd-sourcing procedure in NLI datasets in English too (Parrish et al., 2021). Procedures of data collection using crowd-sourcing have become an increasingly prominent issue in current NLP research, as many fields in NLP rely heavily on crowd-sourced resources (Samuel R. Bowman and Dahl, 2021).

## 5.3 Corpus of Original Chinese Natural Language Inference: OCNLI

In this section, we describe our data collection and annotation procedures. Following the standard definition of NLI (Dagan et al., 2006), our data consists of ordered pairs of sentences, one *premise* sentence and one *hypothesis* sentence, annotated with one of three labels: Entailment, Contradiction, or Neutral (see examples in Table 5.6).

Following the strategy that Williams et al. (2018) established for MNLI, we start by selecting a set of premises from a collection of multi-genre Chinese texts, see Section 5.3.1.

We then elicit hypothesis annotations based on these premises using expert annotators (Section 5.3.2). We develop novel strategies to ensure that we elicit diverse hypotheses. We then describe our verification procedure in Section 5.3.3.

### 5.3.1 Selecting the Premises

Our premises are drawn from the following five text genres: government documents, news, literature, TV talk shows, and telephone conversations. The genres were chosen to ascertain varying degrees of formality, and they were collected from different primary Chinese sources. The government documents are taken from annual Chinese government work reports from 1990 to 2020[2]. The news data are extracted from the news portion of the Lancaster Corpus of Mandarin Chinese which were from 1990s (McEnery and Xiao, 2004). The data in the literature genre are from two contemporary Chinese novels[3], and the TV talk show data and telephone conversations are extracted from transcripts of the talk show *Behind the headlines with Wentao*[4] and the Chinese Callhome transcripts (Wheatley, 1996).

As for pre-processing, annotation symbols in the Callhome transcripts were removed and we limited our premise selection to sentences containing 8 to 50 characters.

### 5.3.2 Hypothesis Generation

As mentioned in chapter 5.1.1, one issue with the existing data collection strategies in MNLI is that humans tend to use the simplest strategies to create the hypotheses, such as negating a sentence to create a contradiction. This can potentially create unrealistic hypotheses. To create more realistic, and thus more challenging data, we propose a new hypothesis elicitation method called *multi-hypothesis* elicitation. We collect a total of four subsets of inference pairs in order to compare the proposed method with the MNLI annotation method. Our first subset asks a single annotator to create an entailed sentence, a

[2] http://www.gov.cn/guowuyuan/baogao.htm, last visited 4/21/2020, same below.
[3] *Ground Covered with Chicken Features* by Liu Zhenyun, *Song of Everlasting Sorrow* by Wang Anyi.
[4] http://phtv.ifeng.com/listpage/677/1/list.shtml.

neutral sentence and a contradictory sentence given a premise (Condition: SINGLE), which is the same as the annotation procedure in MNLI. The other three subsets use some variant of the multi-hypothesis elicitation which we explain now.

**Multi-hypothesis elicitation** In this newly proposed setting, we ask the writer to produce *three* sentences per label, resulting in three entailments, three neutrals and three contradictions for each premise, which forms the data for the second subset: MULTI. I.e. we obtain a total of nine hypotheses if the writer is able to produce that many inferences, which is indeed the case for most premises in our experiment. Our hypothesis is that by asking them to produce three sentences for each type of inference, we "force" them to think beyond the easiest case. We call the 1st, 2nd and 3rd hypothesis by an annotator per label `easy`, `medium` and `hard` respectively, with the assumption that they start with the easiest inferences and then move on to harder ones. First experiments show that MULTI is more challenging than SINGLE, and at the same time, inter-annotator agreement is slightly higher than for SINGLE (see section 5.3.3).

However, we also found that MULTI introduces more hypothesis-only bias. Especially in contradictions, negators such as ¡ ("no/not") stood out as cues, similar to what had been reported in SNLI and MNLI (Gururangan et al., 2018; Pavlick and Kwiatkowski, 2019; Poliak et al., 2018). Therefore we experiment with two additional strategies to control the bias, resulting in another two subsets: MULTIENCOURAGE (*encourage* the annotators to write more diverse hypothesis) and MULTICONSTRAINT (put *constraints* on what they can produce), which will be explained in detail below.

The basis of our instructions are very similar to those for MNLI, but we modified them for each setting:

1. **SINGLE**. We asked the writer to produce one hypothesis per label, same as MNLI[5].

2. **MULTI**. Instructions are the same except that we ask for three hypotheses per infer-

[5]See Appendix B.1 for the complete instructions.

ence type (entailment, neural, contradiction).

3. **MULTIENCOURAGE**. We *encouraged* the writers to write high-quality hypotheses by telling them explicitly which types of data we are looking for, and promised a monetary bonus to those who met our criteria after we examined their hypotheses. Among our criteria are: 1) we are interested in *diverse* ways of making inferences, and 2) we are looking for contradictions that do *not* contain a negator.

4. **MULTICONSTRAINT**. We put *constraints* on hypothesis generation by specifying that *only one out of the three contradictions can contain a negator*, and that we would randomly check the produced hypothesis, with violations of the constraint resulting in lower payment. We also provided extra examples in the instructions to demonstrate contradictions without negators. These examples are drawn from the hypotheses collected from prior data.

We are also aware of other potential biases or heuristics in human-elicited NLI data such as the lexical overlap heuristic (McCoy et al., 2019). Thus in all our instructions, we made explicit to the annotators that no hypothesis should overlap more than 70% with the premise.

Table 5.1 gives a summary of the instructions, the number of total pairs and the mean length of the hypotheses in these four subsets. We observe that the MULTIENCOURAGE and MULTICONSTRAINT conditions have longer hypotheses than SINGLE and MULTI (1–1.5 characters on average), suggesting that our instructions make the annotators produce longer hypotheses. We also notice that in the three MULTI* conditions, we have fewer hypotheses in the `hard` condition, indicating that it is difficult to write the third hypothesis.

**Annotators** We hired 145 undergraduate and graduate students from several top-tier Chinese universities to produce hypotheses. All of the annotators (*writers*) are native speakers of Chinese and are majoring in Chinese or other languages. They were paid roughly 0.3

| Subsets | Instructions | # Pairs / Mean length of hypothesis *H* in characters | | | |
|---|---|---|---|---|---|
| | | Total | `easy` | `medium` | `hard` |
| SINGLE | same as MNLI; one *H* per label | 11,986 / 10.9 | n.a. | n.a. | n.a. |
| MULTI | three *H*s per label | 12,328 / 10.4 | 4,836 / 9.9 | 4,621 / 10.6 | 2,871 / 11.0 |
| MULTIENCOURAGE | MULTI + encouraging annotators to use fewer negators and write more diverse hypotheses | 16,584 / 12.2 | 6,263 / 11.5 | 6,092 / 12.5 | 4,229 / 12.7 |
| MULTICONSTRAINT | MULTI + constraints on the negators used in contradictions | 15,627 / 12.0 | 5,668 / 11.6 | 5,599 / 12.2 | 4,360 / 12.4 |
| total | | 56,486 / 11.5 | | | |

Table 5.1: Overview of the four subsets of data collected. Premises in all subsets are drawn from the same pool of text from five genres. `easy/medium/hard` refers to the 1st/2nd/3rd hypothesis written for the same premise and inference label. Number of pairs in the `hard` condition is smaller because not all premises and all labels have a third hypothesis. See section 5.3.2 for details of the subsets.

RMB (0.042 USD) per P-H pair. No single annotator produced an excessive amount of data to avoid annotator-bias (Geva et al., 2019).[6]

### 5.3.3 Data Verification

Following SNLI and MNLI, we perform data verification to verify that the hypotheses written by the annotators are indeed an entailment/neural/contradictory statement given the premise. This is also referred to as *relabeling* in the NLI literature (Samuel R Bowman et al., 2015; Williams et al., 2018). Concretely, each premise-hypothesis pair is presented to another four independent annotators (*labelers*) to assign one of the three inference labels. Together with the original label assigned by the annotator, each pair has five labels. We then use the majority vote as the gold label. We selected a subset of the writers from the hypothesis generation experiment to be our labelers. For each subset, about 15% of the

[6]One single annotator has completed at most 840 pairs. This is modest compared with the 56k pairs we have in total.

| | SNLI‡ | MNLI‡ | XNLI‡ | OCNLI | | | |
|---|---|---|---|---|---|---|---|
| | | | | SINGLE | MULTI | MULTIENC | MULTICON |
| # pairs in total | 570,152 | 432,702 | 7,500 | 11,986 | 12,328 | 16,584 | 15,627 |
| # pairs relabeled | 56,941 | 40,000 | 7,500 | 1,919 | 1,994 | 3,000 | 3,000 |
| % relabeled | 10.0% | 9.2% | 100.0% | 16.0% | 16.2% | 18.1% | 19.2% |
| 5 labels agree (unanimous) | 58.3% | 58.2% | na | 62.1% | 63.5% | 57.2% | 57.6% |
| 4+ labels agree | na | na | na | 82.2% | 84.8% | 82.0% | 80.8% |
| 3+ labels agree | **98.0%** | **98.2%** | **93.0%** | **98.6%** | **98.8%** | **98.7%** | **98.3%** |
| Individual label gold label | 89.0% | 88.7% | na | 88.1% | 88.9% | 87.0% | 86.7% |
| Individual label author's label | 85.8% | 85.2% | na | 81.8% | 82.3% | 80.2% | 79.7% |
| Gold label author's label | 91.2% | 92.6% | na | 89.8% | 89.6% | 89.6% | 88.2% |
| Gold label author's label | 6.8% | 5.6% | na | 8.8% | 9.2% | 9.0% | 10.1% |
| No gold label (no 3 labels match) | 2.0% | 1.8% | na | 1.4% | 1.2% | 1.3% | 1.7% |

Table 5.2: Results from labeling experiments for the four subsets. MULTIENC: MULTIENCOURAGE; MULTICON: MULTICONSTRAINT. ‡ = numbers for SNLI, MNLI, XNLI are copied from the original papers (Samuel R Bowman et al., 2015; Conneau et al., 2018b; Williams et al., 2018). For XNLI, the numbers are for the English portion of the dataset, which is the only language that has been relabelled.

total data were randomly selected and relabeled. The labelers were paid 0.2 RMB (0.028 USD) for each pair.

**Relabeling results** Our results, shown in Table 5.2, are very close to the numbers reported for SNLI/MNLI: the percentage of pairs that have 3 or more labels in agreement is around 98%. Specifically, the agreement for 5 labels (unanimous) for the SINGLE (62.1%) and MULTI (63.5%) subsets are even higher than SNLI/MNLI ( 58%).

Crucially, the three MULTI* subsets, created using the three variants of the *multi-hypothesis* generation method, have similar agreement to MNLI around 98% for 3 label agreement (see bold numbers on the last three columns of Table 5.3.3, suggesting that producing nine hypotheses for a given premise is feasible. Furthermore, the agreement rates on the `medium` and `hard` portions of the subsets are only slightly lower than on the `easy` portion, with agreement rates of 3 labels at least 97.90% (see bold numbers in the columns with `hard` headings in Table 5.3), suggesting that our data in general is of high quality. Agreement is lower for MULTICONSTRAINT, showing that it may be difficult to produce many hypotheses under these constraints.

| Statistic | MULTI | | | MULTIENCOURAGE | | | MULTICONSTRAINT | | |
|---|---|---|---|---|---|---|---|---|---|
| | easy | medium | hard | easy | medium | hard | easy | medium | hard |
| # pairs relabelled | 668 | 664 | 662 | 1,002 | 999 | 999 | 1,002 | 999 | 999 |
| 5 labels agree (unanimous) | 66.5% | 61.4% | 62.5% | 58.0% | 56.5% | 57.2% | 60.8% | 57.2% | 54.9% |
| 4+ labels agree | 87.0% | 82.1% | 85.2% | 82.2% | 82.6% | 81.2% | 84.5% | 78.6% | 79.4% |
| 3+ labels agree | **99.1**% | **99.1**% | **98.2**% | **98.5**% | **99.1**% | **98.4**% | **98.0**% | **99.0**% | **97.9**% |
| Indiv. label gold label | 90.1% | 88.2% | 88.5% | 87.1% | 87.3% | 86.7% | 87.9% | 86.5% | 85.6% |
| Indiv. label author's label | 84.5% | 80.0% | 82.4% | 80.8% | 80.8% | 78.9% | 82.2% | 79.2% | 77.6% |
| Gold label author's label | 91.5% | 88.1% | 89.3% | 90.4% | 91.4% | 87.1% | 90.1% | 88.3% | 86.1% |
| Gold label author's label | 7.6% | 11.0% | 8.9% | 8.1% | 7.7% | 11.3% | 7.9% | 10.7% | 11.8% |
| No gold label | 0.9% | 0.9% | 1.8% | 1.5% | 0.9% | 1.6% | 2.0% | 1.0% | 2.1% |
| %n_unrelated labels | 0.2% | 0.2% | 0.4% | 0.2% | 0.6% | 0.3% | 0.1% | 0.1% | 0.4% |

Table 5.3: Labeling results for different portions of MULTI, MULTIENCOURAGE and MULTICONSTRAINT.

**Relabeling Results for Different Subsets** In Table 5.3, we present labeler agreement for different portions of MULTI, MULTIENCOURAGE and MULTICONSTRAINT. We observe that the `medium` and `hard` portions in general have lower inter-annotator agreement, but still comparable to SNLI and MNLI. This suggests that writing three hypothese for each label is a feasible and reliable strategy.

### 5.3.4 Relabeling Results for XNLI Development Set

In order to compare the quality of the NLI data in XNLI and our collected data, in a separate relabeling experiment, we compare the quality of human-translated examples from the XNLI dev set and our SINGLE subset. For this experiment, we follow the same procedure as the relabeling experiment in the previous paragraph. We randomly selected 200 examples from XNLI dev, and mixed them with 200 examples from our SINGLE (which has already been verified) for another group of annotators to label. The labelers for these 400 pairs were undergraduate students who did *not* participated in hypothesis generation so as to avoid biasing towards our data.

The results in Table 5.4 show considerably lower agreement: The majority vote of our five annotators only agree with the XNLI gold-label 67% of the time, as compared to the lowest rate of 88.2% on MULTICONSTRAINT. Additionally, 11.6% of the XNLI dev

| Statistic | XNLI dev | SINGLE |
|---|---|---|
| # pairs relabelled (i.e., validated) | 200 | 200 |
| majority label *original* label | **67.0**% | **84.0**% |
| 5 labels agree (excl. "unrelated") | 38.5% | 57.5% |
| 4+ labels agree (excl. "unrelated") | 57.5% | 83.5% |
| 3+ labels agree (excl. "unrelated") | **86.0**% | **98.0**% |
| 5 labels agree | 41.0% | 57.5% |
| 4+labels agree | 62.0% | 83.5% |
| 3+ labels agree | 94.5% | 98.0% |
| majority label = "unrelated" | 8.5% | 0% |
| # individual "unrelated" labels | 125 | 11 |
| # incomprehensible note | 22 | 4 |

Table 5.4: Results for labeling a mixture of 200 pairs of XNLI dev Chinese and 200 pairs of SINGLE, by labelers who did not participated in the hypothesis generation experiment. Note the XNLI dev is translated by crowd translators (Conneau et al., 2018b), not MT systems. The *original* label for XNLI dev Chinese comes with XNLI, which is the same for all 15 languages. The *original* label for SINGLE comes from our relabeling experiments. The results show the poor quality of the sampled XNLI dev examples in Chinese.

examples in Chinese contain more than 10 Roman characters, which are extremely rare in original, every-day Chinese speech/text. These results suggest that XNLI is less suitable as validation set for Chinese NLI, and thus we did not evaluate the models on XNLI dev set in our evaluation.

### 5.3.5 Determining Human Baselines

We follow procedures in Nangia and S. Bowman (2019) to obtain human baselines on OCNLI. In most recent NLU research, a human score is estimated by having a group of human annotators to perform the same task as the models, and their answers are scored against the gold answers in the dataset. The human scores are then assumed to be the upper-bound for this task, and the performance of the models are then compared against this upper-bound (Nangia and S. Bowman, 2019; A. Wang et al., 2019, 2018; Xu et al., 2020). We perform a similar human performance estimation for OCNLI. Specifically, we

| annotator | undergraduate students | PhD students in linguistics |
|---|---|---|
| # pairs annotated | 300 | 300 |
| accuracy (mean agreement w/ OCNLI.test) | **90.3** | **89.3** |
| 5-label agree among annotators | 55.3 | 60.6 |
| 4-label agree among annotators | 82.0 | 83.3 |
| 3-label agree among annotators | 99.3 | 99.0 |
| no majority | 0.7 | 1.0 |

Table 5.5: Human score for 300 randomly sampled examples from the test set of OCNLI.

first prepared 20 training examples from OCNLI.train and instructions similar to those in the relabeling experiment. Then we asked 5 undergraduate students who did *not* participate in any part of our previous experiment to perform the labeling. They were first provided with the instructions as well as the 20 training examples, which they were asked to label after reading the instructions. Then they were given the answers and explanations of the training examples. Finally, they were given a random sample of 300 examples from the OCNLI test set for labeling. We computed the majority label from them, and compare that against the gold label in OCNLI.test to obtain the accuracy. For pairs with no majority label, we use the most frequent label from OCNLI.test (neutral), following Nangia and S. Bowman (2019). Only 0.7% of our examples belong to such cases, as indicated in Table 5.5.

We observe in Table 5.5 that the mean accuracy of the 5 undergraduate students, calculated against the gold labels in the test set of OCNLI, is 90.3%, similar to that of MNLI (92.0%/92.8% for the matched and mismatched portions of MNLI dev set respectively). We performed the same experiment with 5 linguistics PhD students, who are already familiar with the NLI task from their research, and thus their results may be biased. We see a higher 5-label agreement and similar accuracy compared against the gold label of OCNLI.test. We use the score from undergraduate students as our human baseline as it is the "unbiased" score obtained using the same procedure as Nangia and S. Bowman (2019).

| Premise | Genre<br>Level | Majority label<br>All labels | Hypothesis |
|---|---|---|---|
| F / 不 I / 中 ý å , t 个东亚 ‡ ý Ù 个 y ¹ 1 / « C › q Í ˆ ñ<br>But not only China and Japan, the entire East Asian culture has this feature, that is it is deeply influenced by the power. | TV<br>medium | **Entailment**<br>E E E E E | … Ç 两个东亚 ý ¶ Ù 个 y ¹<br>More than two East Asian countries have this feature. |
| Œ „ å 8 ? V S<br>(We need to) perfect our work and trade policies. | GOV<br>easy | **Entailment**<br>E E E E E | 8 ? V S û Ø 不 ³ 之<br>(Our) trade policies still need to be improved. |
| – a † Ì b ù b P „ t { 7 s 也 / 上一代 „ E 事 y ò / Ç e 人了<br>Stories of young couples sitting face-to-face in a cafe is already something from the last generation. She has gone through all that. | LIT<br>medium | **Contradiction**<br>C C C N N | 7 人 Œ s 人 / Ì ù Ì P @ „<br>The man and the woman are sitting back-to-back. |
| 今 ) Ù 一 × 人 s è „ ® È 于 ( â i 举 L<br>Today, this conference which has drawn much attention finally took place in Bonn. | NEWS<br>easy | **Neutral**<br>N N N N C | Ù 一 ® Ÿ š 于 ( ) 举 L<br>This conferences was scheduled to be held yesterday. |
| ï ,今 ) m 们 Ù ? , ï ü .<br>En, it's Saturday today in our place, yeah. | PHONE<br>hard | **Contradiction**<br>C C C C C | ( ) / )<br>It was Sunday yesterday. |

Table 5.6: Examples from our annotated Chinese NLI dataset, one from each of the five text genres. The premise are given to an annotator, and his/her task is to write hypotheses that belong to one of the three categories: entailment, neutral and contradiction. `easy`: 1st hypothesis the annotator wrote for that particular premise and label; `medium`: 2nd hypothesis; `hard`: 3rd hypothesis. **Bold** label shows the majority vote from the annotators.

### 5.3.6 The Resulting Corpus

Overall, we have a corpus of more than 56,000 inference pairs in Chinese. We have randomized the total of 6,000 *relabeled* pairs from MULTIENCOURAGE and MULTICONSTRAINT and used them as the development and test sets, each consisting of 3,000 examples. All pairs from SINGLE and MULTI, plus the remaining 26,211 pairs from MULTIENCOURAGE and MULTICONSTRAINT are used for the training set, about 50,000 pairs. This split ensures that all labels in the development and test sets have been verified, and the number of pairs in the `easy`, `medium` and `hard` portions are roughly the same in both sets. It is also closer to a realistic setting where contradictions without negation are much more likely. Pairs that do not receive a majority label in our relabeling experiment

are marked with "-" as their label, and can thus be excluded if necessary. Examples from our corpus are presented in Table 5.6, with the 5 labels from annotators for each pair.

We note that given the constraints of having equal number of `easy`, `medium` and `hard` examples in dev/test sets, the resulting corpus has high premise overlap between training and dev/test sets, in contrast to the original MNLI design. To ensure that such premise overlap does not bias the current models and inflate performance, we experimented with a smaller **non-overlap** train and test split, which was constructed by filtering parts of the training. This lead to comparable results, despite the non-overlap being much smaller in size, which we detail in chapter 5.4.3. Both the **overlap** and **non-overlap** splits are released for public use at https://github.com/CLUEbenchmark/OCNLI, as well as part of the the public leaderboard at https://www.cluebenchmarks.com/nli.html.

## 5.4 Establishing Baselines for OCNLI

### 5.4.1 Models for Experimentation

To demonstrate the difficulty of our dataset, we establish baselines using several widely-used NLI models tailored to Chinese[7]. This includes the baselines originally used in Williams et al. (2018) such as the continuous bag of words (CBOW) model, the biLSTM encoder model and an implementation of the ESIM (Enhanced Sequential Inference Model) (Q. Chen et al., 2017)[8]. For details of the models, please see chapter 2.2. In each case, we use Chinese character embeddings from S. Li et al. (2018) in place of the original GloVe embeddings for English.

We also experiment with state-of-the-art pre-trained transformers for Chinese (Cui et al., 2019) using the fine-tuning approach from Devlin et al. (2019). Specifically, we use the Chinese versions of BERT-base (Devlin et al., 2019) and RoBERTa-large (Liu et al., 2019) with whole-word masking (see details in Cui et al. (2019)). In both cases, we rely on the

[7] Additional details about all of our models and hyper-parameters can be found in section B.2.
[8] We use a version of the implementations from https://github.com/NYU-MLL/multiNLI.

| Maj. | CBOW | biLSTM | ESIM | BERT | RoBERTa |
|---|---|---|---|---|---|
| 38.1 | 55.7 (0.5) | 59.2 (0.5) | 59.8 (0.4) | 72.2 (0.7) | 78.2 (0.7) |

Table 5.7: Test performance on OCNLI for all baseline models. Majority label is *neutral*. We report the mean accuracy % across five training runs with random re-starts (the standard deviation is shown in parentheses, same below).

publicly-available TensorFlow implementation provided in the CLUE benchmark (Xu et al., 2020)[9]. Following Samuel R. Bowman et al. (2020), we also fine-tune *hypothesis-only* variants of our main models to measure hypothesis-only biases/artifacts.

### 5.4.2 Other Datasets for Experimentation

In addition to experimenting with OCNLI, we also compare the performance of our main models against models fine-tuned on the Chinese training data of XNLI (Conneau et al., 2018b) (an automatically translated version of MNLI), as well as combinations of OCNLI and XNLI. The aim of these experiments is to evaluate the relative advantage of automatically translated data. We also compare both models against the CLUE diagnostic test from Xu et al. (2020), which is a set of 514 NLI problems that was annotated by an independent set of Chinese linguists. Since it is independent from XNLI and OCNLI, it can be used as the out-of-domain data on which we can compare models fine-tuned on XNLI and OCNLI fairly.

### 5.4.3 Baseline Results and Analysis

In this section, we report our main results. Experiments with different neural models in Chinese show that the RoBERTa model outperforms the other ones, but is still about 12 percentage points below human performance (see Table 5.7). Furthermore, models fine-tuned with OCNLI perform better than those fine-tuned with XNLI, evaluated on the CLUE diagnostic dataset. Finally, hypothesis-only biases are unfortunately still high (see Table 5.15),

[9]See: https://github.com/CLUEbenchmark/CLUE

| **Fine-tuning data** / size | | **OCNLI** / 50k | | **XNLI** / 392k | | **Combined** / 443k |
|---|---|---|---|---|---|---|
| Test data | | BERT | RoBERTa | BERT | RoBERTa | RoBERTa |
| OCNLI human | 300 | 90.3* (OCNLI.test) | | | | |
| OCNLI.dev | 3k | 74.5 (0.3) | **78.8** (1.0) | 66.8 (0.5) | 70.5 (1.0) | 76.4 (1.3) |
| OCNLI.test | 3k | 72.2 (0.7) | **78.2** (0.7) | 66.7 (0.3) | 70.4 (1.2) | 75.6 (1.2) |
| CLUE diagnostics | 0.5k | 54.4 (0.9) | 61.3 (1.3) | 53.0 (0.9) | 62.5 (2.9) | **63.7** (2.4) |

Table 5.8: Accuracy on OCNLI, finetuned on OCNLI, XNLI and Combined (50k OCNLI combined with 392k XNLI).

despite OCNLI being a challenging dataset for state-of-the-art neural models.

**How Difficult is OCNLI?** To investigate this, we train/fine-tune all five neural architectures on OCNLI training data and test on the OCNLI test set. The main results for all baseline models fine-tuned on OCNLI are shown in Table 5.7. All of the non-transformer models perform poorly ( 60% accuracy) while BERT and RoBERTa reach a 20 percentage-point advantage over the strongest of these models (ESIM). This shows the relative strength of pre-trained models on our task.

We find that while transformers strongly outperform other baseline models, our best model, based on RoBERTa, is still about 12 points below human performance on our test data (i.e., 78.2% versus 90.3% ). This suggests that models have considerable room for improvement, and provides additional evidence of task difficulty. In comparison, these transformer models reach human-like performance in many of the English GLUE (A. Wang et al., 2018) and SuperGLUE (A. Wang et al., 2019) tasks, for instance the English RoBERTa achieves 88.1% on the GLUE benchmark while the human baseline for GLUE is 87.1%.[10] For NLI tasks in English specifically, the performance of the English RoBERTa on MNLI is 90.4%, and only about 2 percentage-points below the human score (Samuel R. Bowman et al., 2020; Nangia and S. Bowman, 2019), compared to a 12 percentage-points between the human and RoBERTa score on our OCNLI. We see a similar trend for BERT, which is about 18 points below human performance on OCNLI, but the difference is roughly 8

[10]Retrieved from https://gluebenchmark.com/leaderboard, May 10, 2021.

| Test data | BERT | RoBERTa |
|---|---|---|
| OCNLI_dev | 65.3 | 65.7 |
| OCNLI_test | 64.3 | 65.0 |
| OCNLI_test_easy | 63.5 | 64.0 |
| OCNLI_test_medium | 63.9 | 65.6 |
| OCNLI_test_hard | 65.5 | 65.5 |
| MNLI | na. | 62.0 |

Table 5.9: Hypothesis-only baselines for OCNLI (fine-tuned on OCNLI.train) and MNLI (retrieved from Samuel R. Bowman et al. (2020)).

points for MNLI (Devlin et al., 2019). We take these results to indicate that our enhanced annotation procedure indeed produced more challenging data, without sacrificing human agreement on the data. This suggests that having a carefully designed data annotation procedure may be a first step to "fix" the current issues in NLU benchmarks, for instance, the datasets do not contain challenging enough examples and are often too easy for transformer models (Samuel R. Bowman and Dahl, 2021).

We also see much room for improvement on the CLUE diagnostic task, where our best model achieves only 61.3% (a slight improvement over the result reported in Xu et al. (2020)).

**Hypothesis-only biases** We also looked at how OCNLI fares on hypothesis-only tests, where all premises in train and test are replaced by the same non-word, thus forcing the system to make predictions on the hypothesis only. Table 5.9 shows the performance of these models on different portions of OCNLI. These results show that our elicitation gives rise to annotation artifacts in a way similar to most benchmark NLI datasets (OCNLI: 66%; MNLI 62% and SNLI: 69%, as reported in Samuel R. Bowman et al. (2020) and Poliak et al. (2018), respectively). In an unbiased NLI dataset, we would expect hypothesis-only baselines to be around the majority baseline (38% for OCNLI and 33–34% for the English datasets) since no cues in the hypothesis alone should lead to a prediction of any of the inference relations. To determine which words in the hypotheses are cues for model, we

| Word | Label | PMI | Counts |
|---|---|---|---|
| OCNLI | | | |
| 任 U *any* | contradiction | 1.02 | 439/472 |
| 从 e *ever* | contradiction | 0.99 | 229/244 |
| ó *at least* | entailment | 0.92 | 225/254 |
| SINGLE | | | |
| 任 U *any* | contradiction | 0.89 | 87/90 |
| ¡ *no* | contradiction | 0.83 | 582/750 |
| à s *not related* | contradiction | 0.72 | 39/42 |
| MULTI | | | |
| 任 U *any* | contradiction | 0.92 | 97/103 |
| ¡ *no* | contradiction | 0.88 | 721/912 |
| 从 e *ever* | contradiction | 0.75 | 42/46 |
| MULTIENCOURAGE | | | |
| 任 U *any* | contradiction | 0.98 | 198/212 |
| 从 e *ever* | contradiction | 0.96 | 131/137 |
| ó *at least* | entailment | 0.82 | 81/91 |
| MULTICONSTRAINT | | | |
| ó *at least* | entailment | 0.91 | 105/110 |
| ê *only* | contradiction | 0.86 | 179/216 |
| ê *only* | contradiction | 0.77 | 207/280 |

Table 5.10: Top 3 `(Word, Label)` pairs according to PMI for different subsets of OCNLI. "Counts" show (the number of hypotheses with the given `Label` in which the `Word` appears) / (total number of hypotheses in which `Word` appears).

list individual word and label pairs with high pairwise mutual information (PMI), follwoing Samuel R. Bowman et al. (2020) and Poliak et al. (2018). The *(Word, Label)* pair with high PMI means that they have a strong tendency to occur together (see Table 5.10). We found that negative polarity items ("any" 任 U , "ever" 从 e ), negators ("no/not" ¡ ) and "only" ( ê ) are among the indicators for contradictions, whereas "at least" ( ó ) biases towards entailments in OCNLI. We see no negators for the MULTICONSTRAINT subset in the last section of Table 5.10, which shows the effect of putting constraints on the hypotheses that the annotators can produce. Instead, "only" is correlated with contradictions.

**Comparison with XNLI** To ensure that our dataset is not easily solved by simply training on existing machine-translations of MNLI, we show the performance of BERT and RoBERTa when trained on XNLI but tested on OCNLI. The results in Table 5.8 (column XNLI) show a much lower performance than when the systems are trained on OCNLI (70.4% versus 78.2% for RoBERTa), even though XNLI contains 8 times more examples.[11] While these results are not altogether comparable, given that the OCNLI training data was generated from the same data sources and annotated by the same annotators (see Geva et al. (2019)), we still see these results as noteworthy given that XNLI is currently the largest available multi-genre NLI dataset for Chinese. The results are indicative of the limitations of current models fine-tuned solely on translated data. More strikingly, we find that when OCNLI and XNLI are combined for fine-tuning (column Combined in Table 5.8), this improves performance of models fine-tuned with XNLI, but reaches lower accuracies than fine-tuning on the considerably smaller OCNLI (except for the diagnostics[12]). This also suggests that simply adding machine-translated XNLI data to the training data may not help with the performance, even if the translated data are very large in size and the source data of XNLI (which is MNLI) are collected in a very similar manner from OCNLI. It remains to be seen whether using the translated dataset as an auxiliary task in a multi-task learning scenario would be beneficial.[13]

Figure 5.1 shows a learning curve comparing model performance on the independent CLUE diagnostic test, which is the only NLI evaluation dataset independent from XNLI or OCNLI. This is the only NLI data on which a fair comparison can be made at the time of OCNLI creation, since evaluating on OCNLI.dev is somewhat unfair to models fine-tuned on XNLI, but evaluating on XNLI.dev is unfair to OCNLI-fine-tuned models. We plot the learning curve in order to show comparison of the models with different number of training

[11] To ensure that this result is not unique to XNLI, we ran the same experiments using CMNLI, which is an alternative translation of MNLI used in CLUE, and found comparable results.

[12] See the next chapter for a another set of experiments comparing XNLI and OCNLI fine-tuned models on more independent evaluation datasets.

[13] As experimented in the UER model in their CLUE submission: https://github.com/dbiir/UER-py/wiki/CLUE-Classification#ocnli.

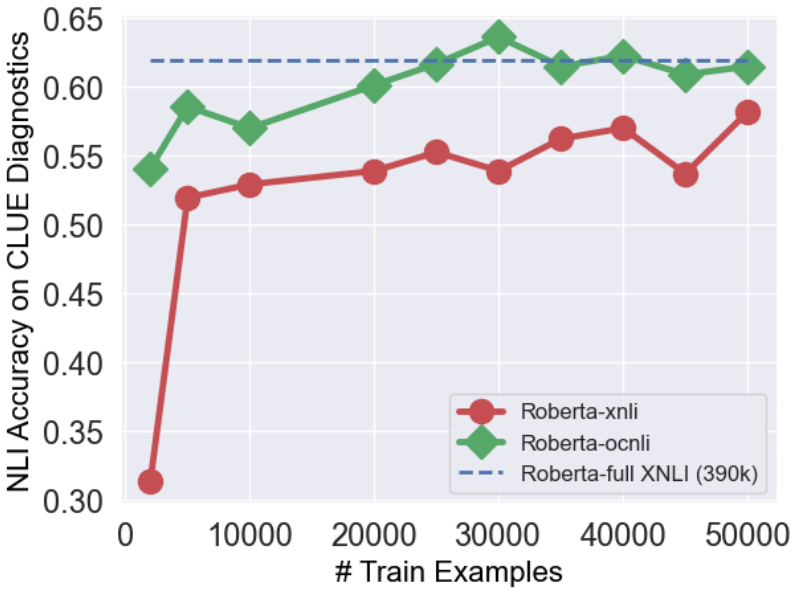



Figure 5.1: Ablation over the number of fine-tuning examples for RoBERTa fine-tuned on OCNLI vs. XNLI.

examples. Here we see that the OCNLI model reaches its highest performance at 30,000 examples while the XNLI model still shows improvements on 50,000 examples. Additionally, OCNLI reaches the same performance as the model finetuned on the full XNLI set, at around 25,000 examples. This provides additional evidence of the importance of having reliable human annotation for NLI data, and the advantage of reliably collected data in a specific language over machine-translated data from other languages. We report more experiments comparing OCNLI, XNLI and OCNLI+XNLI as training data in chapter 6.

**Filtering training data** To mimic the MNLI setting where the training data and the evaluation data (dev/test) have no overlapping premises, we filtered out the pairs in the current training set whose premise can also be found in evaluation. This means the removal of about 20k pairs in OCNLI.train, and results in a new training set which we call OCNLI.train.small, while the development and test sets remain the same. We fine-tune the biLSTM, BERT and RoBERTa models on the new, filtered training data, and the results are presented in Table 5.11.

We observe that our models have a 1.5-2.5% drop in performance when trained with

| Train data / size | OCNLI.train / 50k | | | OCNLI.train.small / 30k | | |
|---|---|---|---|---|---|---|
| | biLSTM | BERT | RoBERTa | biLSTM | BERT | RoBERTa |
| OCNLI.dev | 60.5 (0.4) | 74.5 (0.3) | 78.8 (1.0) | 58.7 (0.3) | 72.6 (0.9) | 77.4 (1.0) |
| OCNLI.test | 59.2 (0.5) | 72.2 (0.7) | 78.2 (0.7) | 57.0 (0.9) | 70.3 (0.9) | 76.4 (1.2) |

Table 5.11: Comparison of models performance fine-tuned on OCNLI.train and OCNLI.train.small. As before, we report the mean accuracy % across five training runs with the standard deviation shown in parenthesis.

the filtered training data. Note that OCNLI.train.small is only 60% of OCNLI.train in size, so we consider this drop to be moderate and expected, and more likely the result of reduced training data, rather than the removal of overlapping premises.

We will release both training sets publicly and also in our public leaderboard (https://www.cluebenchmarks.com/nli.html). We note that similar strategies for controlling dataset size have been used for WINOGRANDE project (Sakaguchi et al., 2020) and their leaderboard (https://leaderboard.allenai.org/winogrande/submissions/public).

## 5.5 Understanding Different Subsets of OCNLI

In order to better understand the differences between the 4 subsets of OCNLI, which reflect the effect of the four annotation strategies, we ran several experiments to answer the following questions:

- **Do the 4 annotation procedures results in differences in the difficulty of the 4 subsets?** We train a RoBERTa model on the training portion of each subset, and test on (a) the dev portion of all the subsets, as well as (b) the roughly 2k expert-crafted, expanded CLUE diagnostics NLI pairs (see section 6.3.2). Results from (a) will give us information as to which subsets are more challenging as dev sets, indicating they can be good evaluation data. Results from (b) will be indicative which subsets are better fine-tuning/training data, since we believe the expanded diagnostics is a comprehensive evaluation set that covers a wide range of linguistic phenomena a model should be able to handle.

| | SINGLE | MULTI | MULTIENC | MULTICON |
|---|---|---|---|---|
| BERT: fine-tune on XNLI | | | | |
| dev_full | 77.3 | 73.6 | 68.6 | 65.8 |
| easy | na. | 74.0 | 70.1 | 68.4 |
| medium | na. | 74.3 | 69.6 | 65.9 |
| hard | na. | 72.5 | 66.2 | 63.1 |
| RoBERTa: fine-tune on XNLI | | | | |
| dev_full | 78.9 | 77.3 | 71.3 | 70.8 |
| easy | na. | 77.2 | 72.8 | 73.5 |
| medium | na. | 78.6 | 71.7 | 70.2 |
| hard | na. | 76.2 | 69.4 | 68.7 |

Table 5.12: Accuracy of XNLI-finetuned models, tested on relabelled parts of different OCNLI subsets.

- **Which subsets minimize hypothesis-only bias?** We train a hypothesis-only model on the training portion of each subset and test on the dev portion of the same subset. The one with highest hypothesis-only accuracy should be the one with largest hypothesis-only bias. This quantitatively analyzes whether the modified annotation procedure increases the hypothesis-only artifacts.

### 5.5.1 Quality of Different Subsets as Evaluation and Training Data

To better understand the effect of having four elicitation methods, we first carried out a set of experiments with XNLI-finetuned models on the different subsets. We used XNLI to avoid imposing specific preferences on the models. Table 5.12 shows a consistent decrease in accuracy from SINGLE through MULTICONSTRAINT, for instance from RoBERTa's accuracy on SINGLE (78.9%) to its accuracy on MULTICON (68.7-73.5%), and a mostly consistent decrease from easy to hard, i.e., about 1-3% drop from easy to medium and about 2% drop from medium to hard (exception: between easy and medium in MULTI). Both trends suggest that *multi-hypothesis* elicitation and improved instructions lead to more challenging elicited data.

| subset | total | dev (verified) | train (before sampling) | train (after sampling) |
|---|---|---|---|---|
| SINGLE | 11,986 | 1,919 | 10,067 | 10,000 |
| MULTI | 12,328 | 1,994 | 10,334 | 10,000 |
| MULTIENCOURAGE | 16,584 | 3,000 | 13,584 | 10,000 |
| MULTICONSTRAINT | 15,627 | 3,000 | 12,627 | 10,000 |

Table 5.13: Number of pairs in each subset for experimentation in chapter 5.5.

| Evaluate on | mean | SINGLE | MULTI | Fine-tuned on MULTIENCOURAGE | MULTICONSTRAINT |
|---|---|---|---|---|---|
| SINGLE | 76.71 | **78.70** | 76.86 | 76.21 | 75.08 |
| MULTI | 76.37 | **77.81** | 77.61 | 75.40 | 74.64 |
| MULTIENCOURAGE | 74.69 | 73.67 | **75.26** | 75.16 | 74.65 |
| MULTICONSTRAINT | 72.80 | 71.70 | **73.46** | 72.80 | 73.22 |
| Expanded diagnostics | - | **64.35** | 63.10 | 62.36 | 60.75 |

Table 5.14: Fine-tuning and evaluating on the subsets of OCNLI, using the Chinese RoBERTa model.

Next, we conduct more experiments where we use the training portion of different subsets as the fine-tuning data and evaluate on the dev portion of the subsets. As a reminder, each of the four subsets has more than 12,000 NLI pairs, 15% out of which have been verified (i.e., received 5 labels) which will be used as the dev portion of each subset. That leaves roughly 10,000 NLI pairs for each subset. We then sampled from these remaining data exactly 10,000 examples for each subset, so that they have equal amount of training data. See the summary in Table 5.13.

We then fine-tune 4 RoBERTa models on the training data of each subset and evaluate on their dev sets, as well as an expanded version of CLUE diagnostics which contain about 2,000 hand-written NLI examples (see chapter 6.3.2 for details). The expanded CLUE diagnostic dataset serves as an out-of-domain evaluation benchmark that evaluates the general NLI ability of the 4 models. Specifically, each model is fine-tuned on the 10,000 sampled training examples for 3 epochs; the learning rate search space is $\{1e\quad 5, 3e\quad 5, 5e\quad 5\}$. In the end, the learning rate for MULTIENCOURAGE model is set to $1e\quad 5$, while other three models are all set to $3e\quad 5$, after experimentation.

The results are shown in Table 5.14. We report the mean accuracy of the 4 models on the dev set of each subset in the second column. As we can see, overall, MULTICONSTRAINT is the most difficult and challenging subset, and SINGLE is the easiest one, indicated by the mean accuracy on these two evaluation sets (72.80% vs. 76.71%).

Another interesting observation is that the difference between SINGLE and MULTI is very small (76.71% vs. 76.37%). Note that SINGLE and MULTI share the same premises per our premise extraction design.[14] This suggests that simply asking annotators to produce more hypotheses per premise may not necessarily create more challenging data. One needs to take extra steps such as those in MULTIENCOURAGE or MULTICONSTRAINT to "force" the annotators to think harder and produce higher-quality inferences (indicated by the lower performance of the models on these data).

If we focus on the column for models fine-tuned on SINGLE, we see a sharp decrease in accuracy from evaluating on SINGLE to the different MULTI subsets. This likely suggests that in the MULTI subsets, there are more diverse inferences which are *not* present in the SINGLE subset. Thus a model fine-tuned on SINGLE finds it difficult to solve the NLI problems in the MULTI subsets.

However, the picture is more nuanced when we examine the effectiveness of the subsets as training data (see the last row in Table 5.14), where the model fine-tuned with SINGLE performs the best on the expanded CLUE diagnostics. This suggests that while the MULTI subsets are more challenging, the models fine-tuned on them may not be as good as the ones fine-tuned with the "vanilla" SINGLE subset. One possibility for this discrepancy is that by design, the MULTI subsets have much fewer unique premises (i.e., less diverse premises), about 1/3 of the SINGLE subset. It is possible that for training a good model, having more unique premise-hypothesis pairs is more important than having more unique hypotheses for a single premise. In other words, given $n$ NLI pairs, it may be better to have

[14] In essence, the premises in MULTI is a subset of those in SINGLE because the annotators are producing three hypotheses per MULTI-premise, but only one hypothesis per SINGLE-premise, and we have roughly the same number of hypothesis-premise pairs in SINGLE and MULTI.

| SINGLE | MULTI | MULTIENCOURAGE | MULTICONSTRAINT |
|---|---|---|---|
| 60.18 | 63.47 | 64.92 | 61.93 |

Table 5.15: Accuracy of hypothesis-only baseline models on different subsets, using Chinese RoBERTa. I.e., we fine-tune RoBERTa on subset $s$ and evaluate also on $s$. Lower accuracy indicates smaller biases.

$\frac{n}{3}$ unique premises where each premise has 3 corresponding hypotheses, rather than having $\frac{n}{9}$ premises but 9 hypotheses per premise.

Our results with regard to the quality as training data of the different subsets corroborate with the findings in two English studies (Samuel R. Bowman et al., 2020; Parrish et al., 2021). Parrish et al. (2021) also show that interventions or modifications in data collection procedure do not yield better training data for NLI. That is, the vanilla MNLI data-collection procedure seems to be the best if the goal is to have high-quality training data.

### 5.5.2 Hypothesis-only Biases in Different Subsets

To understand the hypothesis-only biases produced via different elicitation methods, we fine-tune hypothesis-only models on each subset and evaluate on the dev sets. That is, the training and dev sets for different subsets now consists of only the hypothesis, and the Chinese RoBERTa model is now doing single sentence classification.

The results are shown in Table 5.15. We observe that the `Multi` subsets all have higher hypothesis-only biases. `MultiEncourage` has the largest bias, with the highest accuracy of 65%, whereas `MultiConstraint` has much lower accuracy, suggesting that constraining what the annotators can produce is more effective than encouraging them to write more diverse hypotheses.

To summarize, the MULTI elicitation methods in general produce more challenging NLI pairs, but they could be worse than SINGLE subset as training or fine-tuning data, as indicated by the results on the expanded diagnostics. On the other hand, the MULTI elic-

itation methods tend to create higher hypothesis-only biases, unless the annotators receive explicit constraints on what they can produce as hypotheses. The first point has recently been confirmed in a similar NLI data-collection study in English (Parrish et al., 2021).

## 5.6 Understanding Different Genres of OCNLI

To examine the influence of text genres, we create subsets of the training, dev and test data that contain only the data from one genre. Thus for training, we have five subsets, each reduced to 8,000 examples to ensure that they have equal number of examples: `government`, `literature`, `news`, `phone` and `TV`. We do the same for the dev and test sets.

We fine-tune the Chinese RoBERTa model on each training set and evaluate on the dev set, with results shown in Table 5.16. We first see a clear trend where the same-genre setting produces the best results for all genres except for "tv". Second, the "goverment" genre is the easiest one with a mean of more than 80% accuracy, while the "literature" is the most challenging, with an accuracy that is 12% lower than the "government" genre. In general, we observe that informal genres (literature, phone, and TV) presents a greater challenge than the formal genres (government and news). Finally, on the expanded diagnostics, data from telephone conversations are the worst performing data. This is probably due to the shorter sentence length and the specific language used in these conversations. Models trained on literature, news or TV data perform similarly on the diagnostics, with news data slightly better than the other two, achieving the best score on the expanded diagnostics.

## 5.7 Summary

In this chapter, we presented the first large-scale NLI corpus in Chinese, OCNLI, with more than 56,000 examples. We have demonstrated that requiring the annotators to write more hypotheses given a premise is feasible and will result in more challenging NLI pairs. However, in order to control the hypothesis-only biases, certain constraints need to be

| Evaluate on | mean | Fine-tuned on gov | lit | news | phone | tv |
|---|---|---|---|---|---|---|
| gov | 80.29 | **83.58** | 79.61 | 80.33 | 78.05 | 79.90 |
| lit | 68.05 | 64.17 | **71.55** | 70.08 | 64.51 | 69.92 |
| news | 73.60 | 73.40 | 74.17 | **75.34** | 70.93 | 74.17 |
| phone | 71.40 | 70.97 | 70.89 | 69.32 | **74.33** | 71.51 |
| tv | 69.77 | 67.99 | **72.97** | 68.42 | 67.66 | 71.82 |
| expanded diagnostics | - | 60.49 | 62.00 | **62.58** | 58.93 | 62.33 |

Table 5.16: Accuracy on different genres of `OCNLI.dev`, fine-tuned on different genres of `OCNLI.train`. We see a clear trend which shows that an in-domain (i.e., same-genre) setting produces best results, except for the "tv" genre.

specified for the annotators. Overall, even the best transformer model in Chinese is still far behind human performance (about 12%), suggesting that our dataset is indeed difficult for strong neural models and can facilitate the hill-climbing of these models. (As of Feb 28, 2021, the highest score on the test set on the leaderboard[15] is 83.667%.)

OCNLI also opens several research areas, which we briefly discuss below.

**Reasoning types** The first area is a taxonomy of NLI reasoning types in Chinese. Previous work in English (Joshi et al., 2020; Williams et al., 2020) has used crowd or expert annotators to identify the type of reasoning needed for each NLI problem. These efforts produced annotated dev sets of MNLI and ANLI, where each problem has been assigned one or more reasoning type. For instance, in the taxonomy of in Williams et al., 2020, there are 6 categories at the top level: numerical, basic, reference, tricky, reasoning, and imperfection; at the second level, there are altogether 20 categories: Cardinal/Ordinal numbers, lexical, idioms, coreference, syntactic, pragmatic, plausibility, to name just a few. It will be beneficial to perform a similar type of annotation for OCNLI, which can answer the following questions: 1) what types of reasoning are most typical for Chinese NLI? Are they different from English? 2) what types of reasoning are common in the `easy`, `medium` and `hard` pairs? Do we see a change in the types of reasoning going from `easy` to `hard`? The

[15] https://www.cluebenchmarks.com/nli.html

major difficulty in this direction will be designing a comprehensive and practical taxonomy and annotation guidelines for Chinese.

**Controlling for hypothesis-only biases** As shown in section 5.5.2, there is a fair amount of hypothesis-only bias in OCNLI, despite our best effort to control the biases in the annotation process, with the implementation of different instructions for each subset. Some recent work along the path of OCNLI has shown that iterative improvements on the instructions for multiple rounds guided by linguistic knowledge can produce data that "are more reliably challenging than baseline (MNLI-style data collection)" (Parrish et al., 2021). This is a key issue in NLU at the moment where datasets or benchmarks are easily saturated (Kiela et al., 2021) either because they are not challenging enough or there are too many artifacts that the models can make use of. We hope that OCNLI has shown a viable way for (at least partially) fixing NLU benchmarking (Samuel R. Bowman and Dahl, 2021) and that it can inspire more work that propose better methods for obtaining better datasets in the field of NLU.

# CHAPTER 6

# UNDERSTANDING CROSS-LINGUAL TRANSFER WITH CHINESE NLI

As described in chapter 2.2.2, recent pre-trained multilingual transformer models, such as XLM(-R) (Conneau et al., 2020; Conneau and Lample, 2019), mT5 (Xue et al., 2020) and others (M. Lewis et al., 2020; Liu et al., 2020) have been shown to be successful in NLP tasks for several non-English languages (Choi et al., 2021; Khashabi et al., 2020), as well as in multilingual benchmarks (Artetxe et al., 2020; Conneau et al., 2020; Devlin et al., 2019; Xue et al., 2020). A particular appeal is that they can be used for *cross-lingual* and *zero-shot transfer*. That is, after pre-training on a raw, unaligned corpus consisting of text from many languages, models can be subsequently fine-tuned on a particular task in a resource-rich language (for example, English) and directly applied to the same task in other languages without requiring any additional language-specific training.[1]

Given this recent progress, a natural question arises: does it make sense to invest in large-scale task-specific dataset construction for low-resourced languages, or does cross-lingual transfer alone suffice for many languages and tasks? A closely related question is: how well do multilingual models transfer across specific linguistic and language-specific phenomena?

A common method for understanding the neural models or interpreting the results of them (including the multilingual ones) is to *probe* what the models do/know. The term *probe* usually means to explore the representations of natural language in these models, and this is done either by evaluating models on carefully designed **probing datasets** aimed

[1]This chapter is based on H. Hu et al. (2021). Yixin Nie and I initialized the idea of studying cross-lingual transfer for Chinese NLI, and I proposed the construction of the four datasets. The Chinese HANS is constructed by He Zhou and myself. The stress tests and semantic fragments are constructed by Zuoyu Tian, Yiwen Zhang and myself. The categories in the diagnostic dataset are designed by me; the examples are written by all Chinese linguists on the author list of the paper, including myself. Experiments are run by me. Kyle Richardson helped with the experimental setup and the write-up of the paper.

at specific linguistic phenomena, or **probing classifiers** that examine how well certain linguistic information (e.g., part-of-speech, semantic roles) can be extracted from (certain parts/layers of) the neural model (Tenney et al., 2019a).[2] While there has been much recent work on investigating multilingual models using methods similar to the second probing approach (Karthikeyan et al., 2019; Pires et al., 2019; S. Wu and Dredze, 2019, *inter alia*), a particular limitation is that most studies rely on automatically translated resources such as XNLI (Conneau et al., 2018b) and XQuAD (Artetxe et al., 2020), which makes it difficult to discern the particular linguistic knowledge that is being transferred and the role of large-scale, expert annotated monolingual datasets when building task- and language-specific models.

In this chapter, we investigate the cross-lingual transfer abilities of XLM-R (Conneau et al., 2020) for Chinese natural language inference (NLI) by constructing probing datasets. Our focus on Chinese NLI is motivated by the recent release of the first large-scale, human-annotated Chinese NLI dataset OCNLI (see chapter 5), which we use to directly investigate the role of high-quality task-specific data vs. English-based cross-lingual transfer. To our knowledge, OCNLI is currently the largest non-English NLI dataset that was annotated in the style of English MNLI without any translation. To better understand linguistic transfer, and help benchmark recent state-of-the-art Chinese NLI models, we created 4 categories of probing/adversarial tasks (totaling 17 new datasets) for Chinese that build on several well-established resources for English and the literature on model probing (see Poliak (2020)). Our new resources, which are summarized in Table 6.1, include: a new set of diagnostic tests in the style of the SuperGLUE (A. Wang et al., 2019) and CLUE (Xu et al., 2020) diagnostics; Chinese versions of the HANS dataset (McCoy et al., 2019) and NLI stress tests (Naik et al., 2018), as well as a collection of the basic reasoning and logic *semantic probes* for Chinese based on Richardson et al. (2020). Data and code are available at: https://github.com/huhailinguist/ChineseNLIProbing.

[2] See chapter 2.2.2 for a review of the probing studies.

| dataset | category | n |
|---|---|---|
| Chinese HANS | Lexical overlap | 1,428 |
| | Subsequence | 513 |
| stress tests | Distraction: 2 categories | 8,000 |
| | Antonym | 3,000 |
| | Synonym | 2,000 |
| | Spelling | 11,676 |
| | Numerical reasoning | 8,613 |
| diagnostics | CLUE Xu et al., 2020 | 514 |
| | CLUE expansion (ours) | 796 |
| | World knowledge (ours) | 38 |
| | Classifier (ours) | 139 |
| | Chengyu/idioms (ours) | 251 |
| | Pro-drop (ours) | 198 |
| | Non-core arguments (ours) | 186 |
| semantic probing | Negation | 1,002 |
| | Boolean | 1,002 |
| | Quantifier | 1,002 |
| | Counting | 1,002 |
| | Conditional | 1,002 |
| | Comparative | 1,002 |
| sum | | 43,364 |

Table 6.1: Summary statistics of the four evaluation sets.

**Structure of this chapter** We first review relevant studies on cross-lingual transfer and motivate this work in chapter 6.1, and then present the research questions in chapter 6.2. Next, we describe how we built the four evaluation datasets in chapter 6.3 and the experimental setup in chapter 6.4. Finally the experimental results are presented in chapter 6.5, and the implications for research on multilingual neural models and the limitations of our method are discussed in chapter 6.6.

## 6.1 Motivation for Cross-lingual Linguistic Probing

In this section, we return to the work on (exploring) multilingual transformer models, as well as resources for NLI probing in English, first introduced in chapter 2, and point out the gaps in previous work and motivate our work in this chapter on cross-lingual probing

using the Chinese NLI task.

### 6.1.1 Understanding Multilingual Pre-trained Transformers

Several multilingual pre-trained language models have been proposed, along with their monolingual English version, for instance, mBERT (Devlin et al., 2019), XLM (Lample and Conneau, 2019), XLM-RoBERTa (Conneau et al., 2020), mT5 (Xue et al., 2020), among others. As discussed in chapter 2.2.2, the multilingual transformers can be used in a zero-shot transfer learning scenario, where the model is fine-tuned on labeled data in a high-resource language (English, for instance) and then directly evaluated on the same task in another language (Chinese, for instance) without seeing any labeled fine-tuning data in that language.

Recently, we have also seen a growing number of studies trying to understand multilingual transformers such as XLM-R (Karthikeyan et al., 2019; Nozza et al., 2020; Pires et al., 2019; S. Wu and Dredze, 2019, 2020), as reviewed in chapter 2.2.2. However, the above work either focuses on model architecture, or cross-lingual transfer ability of downstream tasks. Very few studies have examined the transfer ability for different linguistic phenomena. That is, are there specific linguistic phenomena that are easy/difficult to transfer from one language to another? For instance, since most if not all languages have negations (the particular linguistic mechanism for expressing negation may differ), will cross-lingual transfer be relatively successful in such a linguistic category? Will cross-lingual transfer be more difficult for idioms that is unique in one specific language? Thus, in this chapter, our first research question is on cross-lingual transfer learning for different linguistic/logic/reasoning categories, which has received less attention.

### 6.1.2 Lack of Resources for NLI Probing in Chinese

Studies into the linguistic abilities and robustness of current NLI models have proliferated in recent years, partly owing to the discovery of systematic biases, or *annotation arti-*

*facts* (Gururangan et al., 2018; Poliak et al., 2018), in benchmark NLI datasets such as SNLI (Samuel R Bowman et al., 2015) and MNLI (Williams et al., 2018). As we explained in chapter 2.1.4, this has been coupled with the development of new *adversarial tests* such as *HANS* (McCoy et al., 2019) and the NLI *stress tests* (Naik et al., 2018), as well as several new linguistic *challenge datasets* (Geiger et al., 2020; Glockner et al., 2018; Goodwin et al., 2020; Richardson et al., 2020; Saha et al., 2020; Yanaka et al., 2019b, *inter alia*), that focus on a wide range of linguistic and reasoning phenomena. All of this work focuses exclusively on English, whereas we focus on constructing analogous probing datasets tailored to Chinese to help advance research on Chinese NLI and cross-lingual transfer.

Furthermore, there has been a surge in the development of NLI resources for languages other than English, for instance OCNLI (chapter 5) for Chinese, Amirkhani et al. (2020) for Persian, Hayashibe (2020) for Japanese, Fonseca et al. (2016) for Portuguese, among others. One pressing issue at this point is the usefulness of language-specific NLI datasets collected from scratch, in an era where machine-translated data are all too easy to obtain. Is it necessary to build a dataset like OCNLI? Is it good enough to use crowd-translated XNLI, or even an all machine-translated (MT) corpus as experimented in Turkish (Budur et al., 2020), which shows that the MT data has acceptable quality and can be obtained at much lower cost than manual annotation? These concerns motivate the second research question.

## 6.2 Research Questions

In this chapter, we ask the following research questions:

1. How does zero-shot, cross-lingual transfer perform on OCNLI? Can this transfer be successful for various adversarial and probing datasets in Chinese? Specifically, we are comparing the zero-shot performance of multilingual transformer models such as XLM-R against Chinese monolingual models such as the Chinese RoBERTa. Additionally, how does the cross-lingual transfer work in the reverse direction, i.e., trans-

fering from Chinese to English?

2. How does OCNLI compare with XNLI? Specifically, does OCNLI bring performance gains that cannot be achieved with XNLI? This addresses the question of whether it is necessary to build language-specific datasets from scratch, instead of the cheaper method of using machine-translated data directly.

To answer the first question, we perform cross-lingual transfer experiments with XLM-R on multiple NLI corpora we create, including OCNLI. Specifically, we first construct different evaluation datasets for Chinese natural language inference that target various linguistic phenomena. Instead of constructing completely new datasets, we design datasets that are parallel with existing work in English. This allows us to directly compare the transfer learning results in both directions. That is, not only can we fine-tune the models on English NLI datasets and evaluate their zero-shot transfer learning abilities on our constructed Chinese datasets, we can also fine-tune the models on Chinese and perform the same zero-shot transfer evaluation on English, since the evaluation datasets are parallel between the two languages. We further analyze the transfer learning results on specific linguistic phenomena by utilizing the categories defined in our datasets.

To answer the second research question, we compare the models fine-tuned with OCNLI and the models fine-tuned with XNLI. If the former perform better than latter, then we have evidence showing the need for a human annotated NLI dataset for Chinese.

In the next section we will describe in detail the creation of our four evaluation/challenge datasets.

## 6.3 Creating Adversarial and Diagnostic Datasets for Chinese NLI

In this section, we describe the details of the 4 types of challenge datasets we constructed for Chinese to study cross-lingual transfer (summarized in Table 6.1). They fit into two general categories: **Adversarial datasets** (chapter 6.3.1), and **Probing/diagnostic datasets**

(chapter 6.3.2), which are all based on existing datasets in English tailored to Chinese. As explained in chapter 2.1.1, adversarial datasets are those that are designed to expose the weaknesses in the model, i.e., examples that the model systematically return wrong predictions, whereas probing datasets include examples that probe whether a model has certain linguistic knowledge or information in its representation.

While we aim to mimic the annotation protocols pursued in the original English studies, we place the additional methodological constraint that each new dataset is vetted, either through human annotation using a disjoint set of Chinese linguists, or through internal mediation among local Chinese experts; details are provided below.

We choose to create these datasets for the following reasons: 1) they are commonly used in probing English NLI models; 2) they cover a range of representative types of evaluation data: adversarial and probing; 3) they include both hand-written and synthesized data which will allow us to examine model ability more comprehensively.

### 6.3.1 Adversarial datasets

Examples from the 7 adversarial tests we created are illustrated in Table 6.3.[3]

**Chinese HANS** McCoy et al. (2019) discovered systematic biases/heuristics in the MNLI dataset, which they named "lexical/subsequence/constituent" overlap. "Lexical overlap" is defined to be the pairs where the vocabulary of the hypothesis is a subset of the vocabulary of the premise. For example, "*The boss is meeting the client.*" and "*The client is meeting the boss.*", which has an entailment relation. However, lexical overlap does not necessarily mean the premise will always entail the hypothesis, for instance, "*The judge was paid by the actor.*" does not entail "*The actor was paid by the judge.*" (examples from McCoy et al. (2019)). Thus a model relying on the heuristic will fail catastrophically on the second case. The "sub-sequence/constituent overlap" is a special case of "lexical overlap"

[3] A more detailed description of the data creation process can be found in Appendix C.1.

| Heuristic | entailment | contradiction | neutral |
|---|---|---|---|
| lexical overlap | 944 | 155 | 109 |
| subsequence | 190 | 10 | 18 |

Table 6.2: Distribution of the two heuristics in OCNLI

where the overlapping words form a contiguous sub-sequence/syntactic constituent.[4] See chapter 2.1.4 for a description of the English HANS dataset.

Inspired by the English HANS, we first examine whether OCNLI also possesses such biases, as it has a similar annotation procedure as MNLI. We follow the design of the original HANS dataset, and adapt their scripts[5] to extract examples in OCNLI that satisfy the two heuristics.

A total of 1,426 examples are extracted, with their distributions shown in Table 6.2. We find a heavy bias towards "entailment", where 79.5% of such examples are "entailment", similar to MNLI, where the percentage for "entailment" is 87.9%. This indicates that a model fine-tuned on OCNLI is likely to make wrong predictions on contradiction and neutral pairs.

Next, to construct a Chinese HANS dataset, we look at all the NLI pairs extracted from OCNLI, and generalize their patterns into syntactic structures that will be used to generate synthesized NLI pairs conforming to the two heuristics based on a given vocabulary. For example, one contradictory pair that satisfies the lexical overlap heuristics is of the following syntactic structure: "$N_1$ 不 / 不 $V_1$ $N_2$" (*It is not the case that* $N_1$ *does not* $V_1$ $N_2$) CONTRADICTS "$N_1$ 不 $V_1$ $N_2$" ($N_1$ *does not* $V_1$ $N_2$). Once we have such pairs of syntactic structures, we can generate as many contradictory pairs that satisfy the lexical overlap heuristic as possible. We call a pair of syntactic structures a *template*, following the English HANS.

[4] McCoy et al. (2019) focused on these three heuristics as they expect different model architectures may become susceptible to different heuristics. Specifically, the bag-of-words models are most likely to be vulnerable to the lexical overlap heuristic; RNN-based models are likely to be led astray by the sub-sequence heuristic; finally, tree-based models may be prone to the constituent heuristic.

[5] https://github.com/tommccoy1/hans

| | category | n | premise | hypothesis | label |
|---|---|---|---|---|---|
| Chinese HANS | Lexical overlap | 1428 | 们 Š ö L L X Y ( 5 q b 了。We left the bank clerk in the cinema. | ö L L X Š 们 Y ( 5 q b 了。The bank clerk left us in the cinema. | C |
| | Subsequence | 513 | ô ‹ ý / • • Å „。Who told you that all lawyers wear suits. | ‹ ý / • • Å „。All lawyers wear suits. | C |
| stress tests | Distraction (add to premise) | 4000 | ý 业 9 i „ ï Œ ¹ ˆ ? V ò Ï n，且 Z Œ K / ú b „ Å 人不 " g È Ð ¨。The policy of the reform of state-owned enterprises is now clear, and patients who just had surgery shouldn't have intense exercise. | 9 ，不 X ( ý 业。The state-owned enterprises don't exist. | C |
| | Distraction (add to hypothesis) | 4000 | Ù ö N ¶ b P $ á 了 Â Â „ 人。During this time, the Li family's backyard is full of people who came to visit. | Ù 0 ¹ 个 Ó N „ 人 ¶，且 „ 不 / G „。There is a Li family here, and true is not false. | E |
| | Antonym | 3000 | 一些 0 ¹ " ? 6 / Û þ ƒ '。The disagreement about local revenue is relatively big. | 一些 0 ¹ " ? 6 / Û þ ƒ 。The disagreement about local revenue is relatively small. | C |
| | Synonym | 2000 | w è Ä ð ¾ ô 了什么。What can you tell from the difficulties from Kaifu's attempt to set up a cabinet? | w è Ä p ¾ ô 了什么。What can you tell from the hardships from Kaifu's attempt to set up a cabinet? | E |
| | Spelling | 2980 | « 上 ù 一件 å , Ñ „ É ' c , K Ò ( - R Ì。(Someone is) wrapped up in a big cotton coat the factory gave with hands in the sleeves | « 上 ( 一件 c 。There's at least [typo] one coat on the body. | E |
| | Numerical reasoning | 8613 | ¢ Ï Ÿ S 不 0 510个 W。Xiaohong types fewer than 510 words per min. | ¢ Ï Ÿ S 110个 W。Xiaohong types 110 words per min. | N |

Table 6.3: Example NLI pairs in Chinese HANS and stress tests with translations.

In total, we wrote 29 templates for the *lexical overlap* heuristic and 11 templates for *sub-sequence overlap*.[6] Using the templates and a vocabulary of 263 words which are all from the vocabulary of the OCNLI training set,[7] we generated 1,941 NLI pairs. See Table 6.3 for examples.

Next, we describe the 6 categories of stress tests we created, building on the English *stress tests* (Naik et al., 2018), illustrated in Table 6.3. Among the 6 categories, "antonym", "synonym", "spelling" and "numerical reasoning" are directly analogous to the English stress tests. However, our "distraction" category covers three conditions in the English tests: "length mismatch", "word overlap" and "negation". The reason for this deviation is that while Naik et al. (2018) designed these three categories since their error analyses of the

[6]For details of the templates, see Table C.2 and Table C.3 in the Appendix.

[7]The vocabulary is small because we have to satisfy constraints on verb valency and selection criteria, such as "the subject of *eat* must be animate".

NLI system back then on the MNLI dev set show that "length mismatch", "word overlap" and "negation"are sources for error, our goal is different in that we are interested in the effect of adding distractions (tautologies or irrelevant information) to the premise/hypothesis. See chapter 2.1.1 for a description of the English stress tests.

**Distraction** We add distractions either to the premise of the hypothesis (see examples in Table 6.3); the distractions are either tautologies ("true is not false") or a true statement from our world knowledge ("Finland is not a permanent member of the UN security council"), which should not influence the inference label. We control whether the distraction contain a negation or not, and thus create four conditions: *premise-negation*, *premise-no-negation*, *hypothesis-negation*, and *hypothesis-no-negation*. See Table 6.3 for examples and Appendix C.1 for more details.

**Antonym** We replace a word in the premise with its antonym to form a contradiction. To ensure the quality of the resulting NLI pairs, we manually examine the initially generated data and decided to only replace nouns and adjectives, as they are more likely to produce real contradictions.

**Synonym** We replace a word in the premise with its synonym to form an entailment.

**Spelling** We replace one random character in the hypotheses with its homonym (character with the same *pinyin* pronunciation ignoring tones) as this is one of the most common types of misspelling in Chinese.

**Numerical reasoning** We create a probing set for numerical reasoning, following simple heuristics such as the following. When the premise is *Mary types x words per minute*, the entailed hypothesis can be: *Mary types less than y words per minute*, where $x$ $y$. A contradictory hypothesis: *Mary types y words per minute*, where $x$ " $y$ or $x$ ! $y$. Then a neutral pair can be produced by reversing the premise and hypothesis of the above entailment pair.

4 heuristic rules (with 6 words for quantification) are used and the seed sentences are extracted from Ape210k (Zhao et al., 2020), a dataset of Chinese elementary-school math problems. The resulting data contains 8,613 NLI pairs.

For **quality control** and to compute human performance, we randomly sampled 50 examples from all subsets and asked 5 Chinese speakers to verify. Our goal is to mimic the human annotation protocol from Nangia and S. Bowman (2019), which gives us a *conservative* estimate of human performance given that our annotators received very little instructions. Their majority vote agrees with the gold label 90.0% of the time, which suggests that our data is of high quality and allows us to later compare against model performance.[8]

### 6.3.2 Probing/diagnostic dataset

While the Chinese HANS and stress tests are designed to adversarially test the models, we also create probing or diagnostic datasets which are aimed at examining the models' linguistic/logic abilities.

**Hand-crafted diagnostics** We expanded the diagnostic dataset[9] from the Chinese NLU Benchmark (CLUE) (Xu et al., 2020) in the following two ways. See Table 6.4 for examples.

First, 6 Chinese linguists (PhD students) created diagnostics for 4 Chinese-specific linguistic phenomena.

1. *pro-drop*: subjects or objects in Chinese can be dropped when they can be recovered from the context (C. N. Li and Thompson, 1981). Thus the model needs to identify the subject/object from the context.

[8] Specifically: 98.0% on Chinese HANS, 86.0% on the stress tests. For comparison, different subsets of the English stress tests receives 85% to 98% agreement (Naik et al., 2018).

[9] See chapter 2.1.2 for a description of the CLUE diagnostics.

| category | n | premise | hypothesis | label |
|---|---|---|---|---|
| CLUE (Xu et al., 2020) | 514 | 些f œ"(lq¡ Ì1L。<br>Some students like to sing in public showers. | 些s œ"(lq¡ Ì1L。<br>Some female students like to sing in public showers. | N |
| CLUE expansion (ours) | 800 | ÷KÅ*K@ Å†„ŠM (一wM一Cm~个。<br>There are only one thousand six hundred beds in all hotels in Reykjavik. | ÷KÅ*K Å†„ŠM…Ç一C个。<br>Some hotel in Reykjavik has over a thousand beds. | N |
| World Knowledge (ours) | 37 | 上w( 京„W¹。<br>Shanghai is to the south of Beijing. | 京(上w„W¹。<br>Beijing is to the south of Shanghai. | C |
| Classifier (ours) | 138 | Ù些iP 了一个ùœ。<br>These children ate an apple. | Ù些iP 了一Pùœ。<br>These children ate a basket of apples. | N |
| Chengyu/idioms (ours) | 250 | Ù.人ïá>—^J `一个5ÝSÇ»SIÊÇ œ不*¾ó。<br>These people are so cunning! If you call them, you would hit grass alert snake. The consequences would be unimaginable. | `S5ÝÇ» ©Ù.人ßÉ 不}„Óœ。<br>If you call them, it will alert them, and bring negative consequences. | E |
| | | *same as above* | Ù些á>„人{了^ Ç。<br>These cunning people have raised a lot of snakes. | N |
| Pro-drop (ours) | 197 | Á了^ f ÈÙ 们 了两个 ö !•Œ主任È于ï以下í了。<br>After (*pro*) meeting many students and (*pro*) having two hours of meeting with the teachers, the principal and the director can finally get off work. | !•Á了^ f 。<br>The principal met many students. | E |
| | | *same as above* | 们Á了^ f 。<br>The teachers met many students. | N |
| Non-core arguments (ours) | 185 | sö ×Åý"k„ 今)t9SM 了。<br>Zhiyi Fan usually kicks full back (meaning "plays full back in soccer games"), but today he switched to playing forward. | ×ÅÏ8(•"ù¹„ k。<br>Zhiyi Fan usually uses his legs to kick the other team's full back. | N |

Table 6.4: Example NLI pairs in expanded diagnostics with translations.

2. *four-character idioms* (i.e., í *Chengyu*). They are a special type of Chinese idioms that has exactly four characters, usually has a figurative meaning different from the literary meaning of the characters, for instance, SIÊÇ *hit hay startle snake* (behaving carelessly and causing your enemy to become vigilant). We construct examples to test whether models understand the figurative meaning in the idioms. Specifically, we first create a premise $\mathcal{P}$ which includes the idiom. In $\mathcal{P}$, we provide enough context so that there is a strong tendency for a human to interpret the idiom figuratively. Then we create an entailed hypothesis that is based on the figurative interpretation, and a neutral/contradictory hypothesis that uses the literal meaning (see Table 6.4 for an example). For each $\mathcal{P}$ we write 3 hypothesis, one for each inference relation.

3. *classifiers* (or measure word): in Chinese, when modified by a numeral, a noun must be preceeded by a category of words called classifier. They can be semantically vacuous but sometimes also carry semantic content: 一 9 ü *one pi wolf* (one wolf); 一 ¤ ü *one qun wolf* (one pack of wolves). Our examples require the model to understand the semantic content of the classifiers.
4. *non-core arguments*: in Chinese syntax, sometimes a noun phrase at the argument position (e.g., object) is not serving as an object, but rather functions a prepositional phrase: 今 ) w P 不 É P 。 *today eat chopsticks, not eat fork* (We eat **with** chopsticks today, not with fork). Sun (2009) shows that this structure is very productive in Chinese and we take example sentences from her dissertation.

Second, we double the number of diagnostic pairs for all 9 existing linguistic phenomena in CLUE with pairs whose premise are selected from a large Chinese news corpus[10] and hypotheses are hand-written by our linguists, to accompany the 514 artificially created data in CLUE.

For quality control, each pair is double-checked by local Chinese linguists not involved in this study and the controversial cases were discarded after a discussion among the 6 linguists. The resulting new diagnostic dataset is 4 times as large as the original one, with 2,121 NLI pairs.

We made sure that the linguistic phenomena are not merely present in the examples, but a system must have the reasoning skills with respect to the phenomena to make correct predictions. For instance, our examples for Chinese idioms are not merely sentences involving idioms. Rather, knowledge about the idioms is required for correctly making the inference.

**Semantic fragments** Following Richardson et al. (2020) and Salvatore et al. (2019),[11] we design synthesized fragments to examine models' understanding ability of six types of

[10] We use the BCC corpus (Xun et al., 2016): http://bcc.blcu.edu.cn/.
[11] See chapter 2.1.1 for a description of the English semantic fragments.

| category | premise | hypothesis | label |
|---|---|---|---|
| Boolean | U å 、 ¡ — ½ 、 N ý ñ..... ê 0 Ç 临 ~ D ~ ¿ 。<br>*person*$_1$, *person*$_2$ ... have only been to *location*$_1$. | U å ¡ 0 Ç u 义 ¢ ± — : 。<br>*person*$_1$ has not been to *location*$_2$. | E |
| Comparative | ™ ö ä Ô f 书 ~ 、 b ‰..... H · È y ý<br>™ ö ä Œ 亚 „ ³ n 一 7 ' 。*person*$_1$ is younger than *person*$_2$, ..., *person*$_n$; *person*$_1$ is as old as *person*$_m$ | 亚 „ ³ n Ô • ù ' 。*person*$_m$ is older than *person*$_{n\ 2}$. | C |
| Conditional | ..... F – 8 0 Ç c Þ š W ¿ , œ F – 8 ¡ 0 Ç c Þ š W ¿ £ 么 , 0 Ç \| & ? c × 。... *person*$_n$ has been to *location*$_n$. If *person*$_n$ hasn't been to *location*$_n$, then *person*$_m$ has been to *location*$_m$. | , ¡ 0 Ç \| & ? c × 。*person*$_m$ hasn't been to *location*$_m$. | N |
| Counting | é ð Ä ê å ± Ç W ¬ s 、 µ À ¹ ..... u 8 。<br>*person*$_1$ only hugged *person*$_2$, *person*$_3$ ... *person*$_8$. | é ð Ä å ± Ç ... Ç 10个人。<br>*person*$_1$ hugged more than 10 people. | C |
| Negation | " þ ê 0 Ç [ _ » à : 丰 † < ê 0 Ç ' Þ n p – : ..... *person*$_1$ only went to *location*$_1$; *person*$_2$ only went to *location*$_2$; .... | " þ ¡ 0 Ç ' Þ n p – : 。<br>*person*$_1$ has not been to *location*$_2$. | E |
| Quantifier | 人 0 Ç Ï 一个 0 ¹ å ± Ç Ï 一个人。<br>Someone has been to every place and hugged every person. | ‹ s ¡ å ± Ç 一 。*person*$_1$ hasn't hugged *person*$_2$. | N |

Table 6.5: Example NLI pairs for semantic/logic probing with translations. Each label for each category has 2 to 4 templates; we are only showing 1 template for 1 label. 1,000 evaluation examples are generated for each category.

linguistic and logic inference: **boolean**, **comparative**, **conditional**, **counting**, **negation** and **quantifier**, where each category has 2-4 templates. See example templates and NLI pairs in Table 6.5.

The data is generated using context-free grammar rules and a vocabulary of 80,000 person names (Chinese and transliterated), 8659 city names and expanded predicates and comparative relations in Richardson et al. (2020) to make the data more challenging. As a result, we generated 1,000 examples for each fragment. For quality control, each template was checked by 3 linguists/logicians; also 20 examples from each category were checked for correctness by native speakers of Chinese.

### 6.4 Experimental setup

Our main goal is to test whether cross-lingual transfer is successful for the adversarial and probing data we created. Thus we need to compare the best Chinese monolingual models with the best multilingual models.

**Chinese monolingual models** We mainly experimented with two current state-of-the-art transformer models: RoBERTa-large (Liu et al., 2019) and Electra-large-discriminator (K. Clark et al., 2019). We use the Chinese models released from (Cui et al., 2020)[12] implemented in the Huggingface Transformer library (Wolf et al., 2020).

**Multilingual model** We use XLM-RoBERTa-large (Conneau et al., 2020). We choose XLM-R over mT5 (Xue et al., 2020) because XLM-R generally performs better than mT5 under the same model size. For instance, for zero-shot transfer on XNLI, the accuracy of XLM-R with 560 million parameters is 80.9% (Conneau et al., 2020),[13] while the most similar result for mT5 is achieved by mT5-large (81.1%) which has 1.2 billion parameters. Also, XLM-R as a RoBERTa model is most related architecturally to existing Chinese pre-trained models.

**Fine-tuning data for Chinese models & XLM-R** We use the following four datasets to fine-tune the Chinese RoBERTa and XLM-R.

- XNLI: the full Chinese training set in the machine-translated XNLI dataset, with 390k examples (Conneau et al., 2018b). XNLI is reviewed in chapter 2.1.

- XNLI-small: 50k examples from XNLI, the same size as the training data of OCNLI.

[12] We use `hfl/chinese-roberta-wwm-ext-large` from https://github.com/ymcui/Chinese-BERT-wwm and `hfl/chinese-electra-large-discriminator` from https://github.com/ymcui/Chinese-ELECTRA.

[13] We use the results reported in the XLM-R paper (Conneau et al., 2020). Note that in the XTREME paper (J. Hu et al., 2020), the result for XLM-R is reported as 79.2%.

- OCNLI: 50k examples of the OCNLI training set. We use this to measure the effect of the quality of training data; that is, whether it is better to use small, high-quality training data (OCNLI), or large, low-quality MT data (XNLI). OCNLI is described in chapter 5.

- OCNLI + XNLI: a combination of the two training sets, 440k examples.

**Fine-tuning data for XLM-R** To examine cross-lingual transfer, we finetune XLM-R on English NLI data alone and English + Chinese NLI data:

- MNLI: 390k examples from MNLI.train (Williams et al., 2018).

- English all NLI: we combine MNLI (Williams et al., 2018), SNLI (Samuel R Bowman et al., 2015), FeverNLI (Nie et al., 2019; Thorne et al., 2018) with ANLI (Nie et al., 2020a), a total of 1,313k examples. This will be refered to as En-all-NLI in our results.

- OCNLI + English all NLI.

- XNLI + English all NLI. This and the above settings are to examine whether combining Chinese and English fine-tuning data is helpful; if that is the case, adding which Chinese dataset is more helpful: OCNLI or XNLI?

We fine-tune the models on OCNLI-dev. That is, we fine-tune models with different hyperparameter settings (for details see Appendix C.3) and select the hyperparameters based on the performance on OCNLI-dev. We run 5 models on different seeds and report the mean accuracy of the models with the best hyperparameter setting.

**Chinese-to-English transfer** The second part of our first research question is how the cross-lingual zero-shot transfer will work when transferring from a "low-resource" language (Chinese) to a high-resource one (English). We also run the same experiments for Chinese-to-English transfer, i.e., fine-tuning XLM-R with OCNLI and evaluate on the En-

| Model | Fine-tuned on | Acc | Scenario |
|---|---|---|---|
| RoBERTa | zh MT: XNLI-small | 67.44 | monolingual |
| RoBERTa | zh MT: XNLI | 70.29 | monolingual |
| RoBERTa | zh ori: OCNLI | 79.11 | monolingual |
| RoBERTa | zh: OCNLI + XNLI | 78.43 | monolingual |
| XLM-R | zh MT: XNLI | 72.55 | monolingual |
| XLM-R | zh ori: OCNLI | 79.24 | monolingual |
| XLM-R | zh: OCNLI + XNLI | 80.31 | monolingual |
| XLM-R | en: MNLI | 71.98 | zero-shot |
| XLM-R | en: En-all-NLI | 73.73 | zero-shot |
| XLM-R | mix: OCNLI + En-all-NLI | **82.18** | mixed |
| XLM-R | mix: XNLI + En-all-NLI | 74.12 | mixed |

Table 6.6: Results on OCNLI dev. "**Scenario**" indicates whether the model is fine-tuned on Chinese *only* data (**monolingual**), English data (**zero-shot**) or **mixed** English and Chinese data; results in gray show best performance for each scenario. Best overall result in **bold**. Same below.

glish data: MNLI_dev, English HANS, stress tests, GLUE diagnostics and semantic probing. We find that transferring from OCNLI to English does not perform as well as monolingual English models, likely due to the small size of OCNLI, which has 50k examples, compared to English all NLI, which has 1,313k examples.[14]

## 6.5 Results and discussion

### 6.5.1 Results on OCNLI_dev

Results on the dev set of OCNLI are presented in Table 6.6. For monolingual RoBERTa, the performance is 79.11% in accuracy, similar to that reported in Table 5.8 (78.8%) in chapter 5 on OCNLI.[15] The monolingual Electra achieves a very close accuracy of 79.02%. As we see the same trend in the following experiments, we will therefore only report results on RoBERTa. The machine-translated XNLI falls behind about 9 percentage points.

[14] More results are reported in Appendix C.4.

[15] The difference may be attributed to 2 reasons: 1) the random seeds may be different, 2) the results in chapter 5 are obtained using a Tensorflow implementation, while results in this chapter is obtained using the Transformer library which uses PyTorch (Paszke et al., 2019).

For XLM-R, fine-tuning on MNLI or En-all-NLI gives us reasonable results of around 72% to 74%, which is better than models fine-tuned on XNLI, indicating that fine-tuning on an English data (MNLI) alone can outperform monolingual models fine-tuned on the same data but machine-translated into Chinese (XNLI).[16] This suggests that for an NLU task where high-quality English data exists, when working with multilingual models such as XLM-R, it is worth trying to fine-tune the model on the English data first, rather than machine-translate the data into the language one is working on.

What is also interesting is that combining OCNLI (50k Chinese examples) and En-all-NLI (that is, 1,212k English examples, as described in chapter 6.4) gives us a boost of 2% to 82.18% (a result that surpasses the current published state-of-the-art), showing the power of mixing high-quality English and Chinese training data. This indicates that beyond zero-shot, cross-lingual transfer, multilingual models offer another effective training strategy, that is mixing high-quality data of the same task from different languages.

### 6.5.2 Results on Chinese HANS

Table 6.7 shows results of the Chinese HANS data tested on the aforementioned monolingual models and cross-lingual model.

**Cross-lingual transfer achieves strong results** We first notice that when XLM-R is fine-tuned solely on the English data (En-all-NLI), the performance ( 69%) is only slightly worse than the best monolingual model ( 71%). This suggests that cross-lingual transfer from English to Chinese is quite successful for an adversarial dataset like HANS. Second, adding OCNLI to En-all-NLI in the training data gives a large boost of about 9%, and achieves the overall best result. This is about 12% higher than combining XNLI and the English data, demonstrating the advantage of the expert-annotated OCNLI over machine translated XNLI, even though the latter is about 8 times the size of the former. Despite these

[16]For these experiments we also tested with another Chinese machine-translated MNLI (CMNLI), translated by a different MT system, which was released by CLUE (https://github.com/CLUEbenchmark/CLUE), and obtained similar results.

| Model | Fine-tuned on | Overall | Lexical Overlap | Sub-sequence | Entailment | Non-Entailment | Δ |
|---|---|---|---|---|---|---|---|
| RoBERTa | zh MT: XNLI-small | 49.48 | 58.12 | 25.42 | 99.22 | 30.26 | 37.18 |
| RoBERTa | zh MT: XNLI | 60.80 | 68.99 | 38.01 | 99.74 | 45.76 | 24.53 |
| RoBERTa | zh ori: OCNLI | 71.72 | 75.39 | 61.48 | 99.67 | 60.91 | 18.20 |
| RoBERTa | zh: OCNLI+XNLI | 69.33 | 74.73 | 54.27 | 99.89 | 57.51 | 20.92 |
| XLM-R | zh MT: XNLI | 57.74 | 66.47 | 33.45 | 99.96 | 41.42 | 31.13 |
| XLM-R | zh ori: OCNLI | 61.82 | 65.83 | 50.68 | 99.89 | 47.11 | 32.13 |
| XLM-R | zh: OCNLI+XNLI | 70.31 | 74.25 | 59.34 | 100.00 | 58.84 | 21.47 |
| XLM-R | en: En-all-NLI | 69.56 | 77.62 | 47.13 | 100.00 | 57.80 | 15.93 |
| XLM-R | en: MNLI | 66.74 | 73.12 | 48.97 | 100.00 | 53.89 | 18.09 |
| XLM-R | mix: OCNLI+En-all-NLI | **78.82** | **81.57** | **71.15** | **100.00** | **70.63** | 11.55 |
| XLM-R | mix: XNLI+En-all-NLI | 66.90 | 76.25 | 40.90 | 99.93 | 41.89 | 32.23 |
| Human | | 98.00 | | | | | |

Table 6.7: Accuracy on Chinese HANS. Δ indicates the Δ of accuracy between OCNLI dev and Non-Entailment.

results, however, we note that all models continue to perform below human performance, suggesting more room for improvement.

**Discussion of different heuristics** Our results suggest that examples involving the *sub-sequence* heuristics are more difficult than those targeting the *lexical overlap* heuristics (results in the "sub-sequence" column are much lower than those in the "lexical overlap" column in Table 6.7). We also see a much wider gap between the monolingual model and XLM-R for the sub-sequence category: results from monolingual model are 12% higher than those from XLM-R under the zero-shot transfer setting (61.48% versus 48.79% in "sub-sequence" column in Table 6.7). For the lexical overlap heuristic, however, these two types of model have similar performance (75.39% versus 77.62% in "lexical overlap" column in Table 6.7). Recall that the sub-sequence category includes challenging examples such as: *who told you X* ÷ (does not entail) *X*, or *if X, then Y* ÷ *X*[17], where the hypothesis is a sub-sequence of the premise and the model is prone to mistakenly predict ENTAIL. That is, the examples targeting the sub-sequence heuristic are more difficult than the ones targeting lexical overlap, where the vocabulary of the hypothesis is a subset of the vocabulary of the premise, for instance *John likes Mary* ÷ *Mary likes John*. This suggests that as the difficulty level increases (from lexical overlap to sub-sequence), the performance of the

[17]For instance, *If I know who did it, I will tell you* ÷ *I know who did it.*

zero-shot cross-lingual may quickly fall behind the performance of monolingual models.

**Manual Error Analysis for XLM-R** We manually investigate the examples that the zero-shot cross-lingual XLM-R model (fine-tuned with En-all-NLI) fails on but can be correctly predicted by XLM-R fine-tuned with OCNLI + En-all-NLI. We find that adding OCNLI to all English NLI data, the model performs better in: (I) understanding adverbs indicating possibility such as ï ý *possible*, ¸ *perhaps* and ' , *probably*; (II) predicting "non-entailment" for examples in the "subsequence" heuristic, e.g., (a) ô ‹ ý / • • Å „ 。 *Who told you that all lawyers wear suits.* ÷ ‹ ý / • • Å „ 。 *All lawyers wear suits.* (b) 三 t ž ° Ï 个 Q ý 5 q b 。 *In three years, the goal will be realized that every village has a cinema.* ÷ Ï 个 Q ý 5 q b 。 *Every village has a cinema.* This corroborates with findings in the previous paragraph that having high quality Chinese data in the training set will make the model more robust under more difficult adversarial examples such as those targeting the sub-sequence heuristic.

### 6.5.3 Results on stress tests

Table 6.8 presents the accuracies on all the stress tests. We first see that cross-lingual zero-shot transfer using all English NLI data performs even better than the best monolingual model ( 74% vs. 71% in the "overall" column of Table 6.8). This demonstrates the strong performance of the cross-lingual transfer-learning. Adding OCNLI to all English NLI gives another increase of about 3 percentage points (to 77%), while adding XNLI hurts the performance, again showing the importance of having expert-annotated language-specific data.

**Antonyms and Synonyms** All models except those fine-tuned on OCNLI achieved almost perfect score on the synonym test. However, for antonyms, both mono- and multilingual models fine-tuned with OCNLI perform better than XNLI ( 71% vs. 52-55%). XLM-R fine-tuned with only English NLI data again outperforms the best of monolingual

| Model | Fine-tuned on | Overall | Ant. | Syn. | Distr H | Distr H-n | Distr P | Distr P-n | Spell-ing | num. |
|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | zh MT: XNLI-small | 59.41 | 43.38 | 99.64 | 51.61 | 51.41 | 70.66 | 71.19 | 69.93 | 28.70 |
| RoBERTa | zh MT: XNLI | 66.22 | 52.28 | 99.79 | 54.83 | 53.8 | 74.55 | 74.57 | 72.22 | 53.53 |
| RoBERTa | zh ori: OCNLI | 64.49 | 71.81 | 73.66 | 52.95 | 51.8 | 73.43 | 73.86 | 71.79 | 54.16 |
| RoBERTa | zh: OCNLI + XNLI | 71.01 | 59.39 | 99.06 | 55.87 | 54.64 | 76.83 | 76.50 | 75.48 | 70.18 |
| XLM-R | zh MT: XNLI | 66.87 | 55.53 | **99.96** | 56.11 | 55.29 | 77.69 | 77.9 | 74.37 | 46.81 |
| XLM-R | zh ori: OCNLI | 69.08 | 71.29 | 88.63 | 55.93 | 55.05 | 76.84 | 77.00 | 71.42 | 65.51 |
| XLM-R | zh: OCNLI + XNLI | 71.49 | 61.85 | 99.45 | 58.15 | 57.92 | 79.16 | 79.28 | **77.93** | 61.88 |
| XLM-R | en:MNLI | 67.94 | 65.77 | 99.2 | 55.14 | 54.6 | 75.75 | 75.76 | 70.76 | 50.90 |
| XLM-R | en: En-all-NLI | 74.52 | 80.36 | 97.58 | 54.74 | 53.56 | 73.96 | 73.92 | 71.02 | 82.73 |
| XLM-R | mix: OCNLI + En-all-NLI | **77.36** | **81.93** | 95.09 | **59.23** | **58.00** | **79.88** | **79.92** | 74.53 | **87.77** |
| XLM-R | mix: XNLI + En-all-NLI | 73.57 | 66.15 | 99.68 | 57.02 | 55.51 | 78.38 | 78.53 | 75.15 | 80.33 |
| Human | | 85.00 | 85.00 | 98.00 | 83.00 | 83.00 | 83.00 | 83.00 | 78.00 | 98.00 |

Table 6.8: Accuracy on the stress test. Distr H/P(-n): distraction in Hypothesis/Premise (with negation).

models ( 80% vs. 72%).

Interestingly, adding XNLI to all English NLI data has a large negative impact on the accuracy (a 14% drop), while adding OCNLI to the same English data improves the result slightly (a 1.6% increase).

As antonyms are much harder to learn (Glockner et al., 2018), we take our results to mean that either expert-annotated data for Chinese or a huge English NLI dataset is needed for a model to learn decent representations about antonyms, as indicated by the high performance of RoBERTa fine-tuned with OCNLI (71.81%), and XLM-R fine-tuned with En-all-NLI (80.36%), on antonyms. That is, using machine-translated XNLI will not work well for learning antonyms ( 55% accuracy). On the other hand, for learning synonyms, it is better to use a large-scale dataset, rather than a small but high-quality dataset, as indicated by the low performance of models fine-tuned with OCNLI (73.66%).

**Distraction** Results in Table 6.8 show that adding the distractions to the hypotheses has much more of a negative impact on models' performance, compared with appending distractions to the premises. The difference is about 20% for all models (see "Distr H" column and "Distr P" column in Table 6.8), which has not been reported in previous studies, to the

best of our knowledge.[18] Including a negation in the hypothesis makes it even more challenging, as we see another one percent drop in the accuracy for all models (see "Distr H" column and "Distr H-n" column in Table 6.8). This is expected as previous literature has demonstrated the key role negation plays in the hypothesis (Gururangan et al., 2018; Poliak et al., 2018).

**Spelling** This is one of the few cases where cross-lingual transfer with English data alone falls behind monolingual Chinese models (by a considerable difference of about 4%). Also the best results are based on fine-tuning XLM-R with OCNLI + XNLI, rather than a combination of English and Chinese data. Considering the data is created by swapping Chinese characters with others of the same pronunciation, we take it to suggest that monolingual models are still better at picking up the misspellings or learning the connections between characters at the phonological level.

**Numerical Reasoning** Results in the last column of Table 6.8 suggest a similar pattern: using all English NLI data (more than 1,000k examples) for cross-lingual transfer outperforms the best monolingual model. However, fine-tuning a monolingual model with the small OCNLI (50k examples, accuracy: 54%) achieves better accuracy than using a much larger MNLI (390k examples, accuracy: 51%) for cross-lingual transfer, although both are worse than XLM-R fine-tuned with all English NLI which has more than 1,000k examples (accuracy: 83%). This suggests that there are cases where a monolingual setting (RoBERTa with OCNLI) is competitive against zero-shot transfer with a large English dataset (XLM-R with MNLI). However, that competitiveness may disappear when the English dataset grows to an order of magnitude larger in size or becomes more diverse (recall that En-all-NLI is composed of several different English NLI datasets).

[18] Naik et al. (2018) has conditions that add distractions to the premise and hypothesis respectively. However, the conditions were designed for other purposes and they did not add the same distractions to the premise and the hypothesis. Thus we cannot perform such a comparison.

| model | finetune on | overall | Classifier | Idioms | Non-core arguement | Pro-drop | Time of event |
|---|---|---|---|---|---|---|---|
| RoBERTa | XNLI-small | 62.9 | 65.8 | 64.7 | 55.2 | 80.5 | 60.0 |
| RoBERTa | XNLI | 67.7 | 67.6 | 66.2 | 59.4 | 82.3 | 65.1 |
| RoBERTa | OCNLI | 67.8 | 62.0 | 68.0 | 59.4 | 80.7 | 77.5 |
| RoBERTa | OCNLI + XNLI | 69.3 | 66.3 | 67.1 | 58.6 | 83.0 | 74.0 |
| XLM-R | XNLI | 60.9 | 61.2 | 62.3 | 50.4 | 71.9 | 59.7 |
| XLM-R | OCNLI | 68.0 | 57.6 | 70.1 | 58.0 | 79.6 | 76.3 |
| XLM-R | OCNLI + XNLI | 71.5 | 70.4 | 71.6 | 57.5 | 84.6 | 77.8 |
| XLM-R | MNLI | 70.2 | 70.1 | 73.9 | 57.5 | 86.4 | 70.8 |
| XLM-R | En-all-NLI | 71.9 | 71.8 | **74.3** | 56.2 | 87.4 | 75.7 |
| XLM-R | OCNLI + En-all-NLI | **74.9** | **72.7** | **74.3** | 60.1 | **88.5** | **84.5** |
| XLM-R | XNLI + En-all-NLI | 71.4 | 70.2 | 58.5 | **85.5** | 71.3 | 75.2 |

| model | finetune on | Anaphora | Argument structure | Common sense | Comparatives | Double negation | Lexical semantics | Monotonicity | Negation | World knowledge |
|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | XNLI-small | 59.6 | 67.4 | 54.3 | 61.4 | 48.3 | 60.9 | 59.7 | 66.2 | 39.0 |
| RoBERTa | XNLI | 69.9 | 72.0 | 56.8 | 70.4 | 64.2 | 67.5 | 61.7 | 72.9 | 52.1 |
| RoBERTa | OCNLI | 70.3 | 70.0 | 56.0 | 66.6 | 64.2 | 68.4 | 61.7 | 72.4 | 57.9 |
| RoBERTa | OCNLI + XNLI | 70.1 | 73.5 | 54.9 | 74.1 | 67.5 | 69.1 | 62.5 | 76.0 | 60.0 |
| XLM-R | XNLI | 60.3 | 63.3 | 51.7 | 65.2 | 54.9 | 61.0 | 53.5 | 66.9 | 58.3 |
| XLM-R | OCNLI | 67.4 | 70.3 | 55.3 | 69.8 | 75.8 | 71.1 | 62.5 | 71.1 | 62.1 |
| XLM-R | OCNLI + XNLI | 74.5 | 74.7 | 55.3 | 75.5 | 76.7 | 72.8 | 62.7 | 76.3 | 65.3 |
| XLM-R | MNLI | 69.3 | 72.9 | 48.9 | 76.0 | 62.5 | 67.8 | 62.6 | 77.0 | 62.1 |
| XLM-R | En-all-NLI | 74.9 | 74.8 | 49.1 | 80.5 | 70.8 | 69.1 | 63.8 | 77.8 | 64.2 |
| XLM-R | OCNLI + En-all-NLI | **77.3** | **78.1** | 56.6 | **81.3** | **79.2** | **77.2** | 65.6 | **78.0** | 67.9 |
| XLM-R | XNLI + En-all-NLI | 75.5 | 55.1 | **79.2** | 70.0 | 69.1 | 62.4 | **76.2** | 72.1 | **71.3** |

Table 6.9: Accuracy on the expanded diagnostics. Uniquely Chinese linguistic features at the top, others at the bottom.

### 6.5.4 Results on hand-written diagnostics

Results on the expanded diagnostics are presented in Table 6.9. We first see that XLM-R fine-tuned with only English performs very well, at 70.2-71.9%, even slightly higher than the best monolingual Chinese models (69.3%).

Most surprisingly, in 3/5 categories with uniquely Chinese linguistic features, zero-shot transfer outperforms monolingual models. Only in "non-core arguments" (from expanded diagnostics) and "time of event" (from original diagnostics) do we see higher performance of OCNLI as the fine-tuning data. What is particularly striking is that for "idioms

| model | finetune on | overall | boolean | comparative | conditional | counting | negation | quantifier |
|---|---|---|---|---|---|---|---|---|
| RoBERTa | zh MT: XNLI-small | 46.57 | 32.81 | 34.41 | 61.48 | 81.82 | 33.27 | 35.63 |
| RoBERTa | zh MT: XNLI | 50.64 | 33.35 | 39.02 | 66.55 | 84.51 | 40.92 | 39.50 |
| RoBERTa | zh ori: OCNLI | 47.53 | 35.81 | 34.81 | 62.87 | 69.64 | 49.84 | 32.24 |
| RoBERTa | zh: OCNLI + XNLI | 51.13 | 38.16 | 37.98 | 66.19 | 75.73 | 53.31 | 35.43 |
| XLM-R | zh ori: OCNLI | 54.33 | **54.19** | **49.02** | 52.46 | 79.70 | 59.52 | 31.08 |
| XLM-R | zh MT: XNLI | 50.79 | 33.39 | 35.33 | 66.01 | 87.23 | 33.17 | **49.60** |
| XLM-R | zh: OCNLI + XNLI | 52.43 | 34.51 | 36.93 | 59.98 | 88.70 | 54.37 | 40.08 |
| XLM-R | en: MNLI | 49.09 | 33.27 | 37.98 | 66.25 | 89.70 | 34.69 | 32.65 |
| XLM-R | en: En-all-NLI | 55.37 | 33.43 | 39.70 | 66.65 | 92.34 | 64.11 | 35.99 |
| XLM-R | mix: OCNLI + En-all-NLI | **57.95** | 40.70 | 44.49 | 63.67 | 91.54 | **74.47** | 32.81 |
| XLM-R | mix: XNLI + En-all-NLI | 57.73 | 40.30 | 37.82 | **66.67** | **93.19** | 61.52 | 46.87 |

Table 6.10: Accuracy on the Chinese semantic probing datasets, designed following Richardson et al. (2020).

(*Chengyu*)", XLM-R fine-tuned only on English data even achieves the best result, suggesting that the cross-lingual transfer is capable of learning some meaning representation beyond the surface lexical information, at least for many of the idioms we tested, judging from the accuracy of 74.3%. The overall results indicate that cross-lingual transfer is very successful in most cases. We perform an error analysis on the diagnostics for idioms in chapter 6.6.

Looking at OCNLI and XNLI, we observe that they perform similarly when fine-tuned on monolingual RoBERTa. However, when coupled with English data to be used with XLM-R, we see again a clear advantage of the expert-annotated OCNLI + En-all-NLI, with an accuracy 3 percent higher than XNLI + En-all-NLI.

### 6.5.5 Results on semantic fragments

Results on the semantic probing datasets (shown in Table 6.10) are more mixed. First, the results are in general much worse than the other evaluation data, but overall, XLM-R fine-tuned with OCNLI and all English data still performs the best. The overall lower performance is likely due to the longer length of premises and hypotheses in the semantic probing datasets, compared with the other three evaluation sets. Second, zero-shot transfer is better or on par with monolingual Chinese RoBERTa in 4/6 semantic fragments (except Boolean and quantifier). Third, for Boolean and comparative, XLM-R fine-tuned with

OCNLI has a much better result than all other monolingual models and XLM-R fine-tuned with mixed data.

Turning to the fragments now, we first observe that the models have highest performance on the counting fragment (up to 90%+ accuracy). Note that none of the models have seen any synthesized data from the fragments during fine-tuning. That is, all the knowledge come from the pre-training and fine-tuning on existing NLI datasets. Therefore, for the XLM-R model fine-tuned on En-all-NLI, we can say this is zero-shot in two ways: one is cross-lingual zero-shot in that the model has not seen labeled Chinese data, the other is zero-shot from general NLI data (En-all-NLI) to evaluation data on a very specific reasoning skill (counting), i.e., the model sees no "counting" NLI data. The surprisingly good performance of XLM-R (w/ En-all-NLI) model (92.34%) suggests that it may have already acquired a mapping from counting the words/names to numbers. In the study on English fragments (Richardson et al., 2020), zero-shot transfer learning with English BERT$_{base}$ only achieves about 70% accuracy, much lower than our results with XLM-RoBERTa, showing the size of improvement one may obtain from switching to a larger model.

Nevertheless, for the boolean, comparative and quantifier fragments, most models are still performing at chance level (30-40% in accuracy in Table 6.10), indicating that our logical templates for these fragments are still hard for the models if they are only fine-tuned on general NLI datasets.

### 6.5.6 Results on XNLI_dev

Although we have shown in chapter 5.3.4 that XNLI_dev is problematic because of the translation quality, we still run several experiments on XNLI_dev to compare the results of XLM-R model on different fine-tuning sets, as XNLI is widely used in the field of NLI for benchmarking the models.

We first see that for the monolingual RoBERTa (in the upper half of Table 6.11), models fine-tuned on OCNLI perform much worse than those fine-tuned on XNLI, showing that in-

| model | fine-tuned on | acc |
|---|---|---|
| RoBERTa | zh MT: XNLI-small | 76.85 |
| RoBERTa | zh MT: XNLI | 79.99 |
| RoBERTa | zh MT: CMNLI | 80.37 |
| RoBERTa | zh ori: OCNLI | 71.12 |
| RoBERTa | zh: OCNLI + XNLI | 80.66 |
| XLM-R | zh MT: XNLI | 82.91 |
| XLM-R | zh ori: OCNLI | 75.46 |
| XLM-R | zh: OCNLI + XNLI | 83.31 |
| XLM-R | en: MNLI | 78.97 |
| XLM-R | en: En-all-NLI | 79.43 |
| XLM-R | mix: OCNLI + En-all-NLI | 79.60 |
| XLM-R | mix: XNLI + En-all-NLI | **83.60** |

Table 6.11: Results on XNLI dev. Best results for XLM-R in **bold**, for RoBERTa in *italics*.

domain training generally results in better performance. We also experiment with CMNLI, which is machine translated from the same English MNLI dataset as XNLI, but using a different MT system than the Facebook in-house MT system used for XNLI. We see that fine-tuning with CMNLI or XNLI results in similar accuracy, suggesting that the effect of MT system is minimal for our case.

When we move to the lower half of Table 6.11, we first observe that even with the same Chinese data, XLM-R outperforms the monolingual RoBERTa. For example, the accuracy of finetuning on XNLI is 82.91% (on XLM-R) vs. 79.99% (on RoBERTa). Next, fine-tuning on MNLI alone produces very good results, only 4 percentage points lower than fine-tuning on its machine-translated Chinese version (XNLI). This suggests again that XLM-R is indeed learning some useful meaning representation that works reasonably well even if the fine-tuning data and the evaluation data are not in the same language. Finally, the best performing system this time is XNLI + En-all-NLI, which is not surprising since we are evaluating on the dev set of XNLI.

## 6.6 Discussion

To summarize our findings, we first find that cross-lingual models trained exclusively on English NLI do transfer relatively well across our new Chinese tasks. Specifically, in 3/4 of the challenge datasets we created (stress tests, hand-written diagnostics and semantic fragments), they perform as well or better than the best monolingual Chinese models. A particularly striking result is that such models even perform well on 3/5 uniquely Chinese linguistic phenomena such as *idioms, pro drop*, providing evidence that many language-specific phenomena do indeed transfer. These results, however, come with important caveats: on several phenomena (sub-sequence heuristics, distraction in hypothesis, non-core arguments, etc.) we find that all models continue to struggle and are far outpaced by estimates of human performance, highlighting the need for more language-specific diagnostics tests and models that are robust under challenging settings.

Also, fine-tuning models on mixtures of English NLI data plus high-quality monolingual data (OCNLI) consistently performs the best, whereas mixing with training data automatically translated from English to Chinese (XNLI-zh) can greatly hinder model performance. This shows that high-quality monolingual datasets still play an important role when building cross-lingual models. However, how much benefit a monolingual dataset can bring varies with regard to the type of evaluation data we are using. For instance, from the results of the expanded diagnostics in Table 6.9, we see that adding OCNLI to En-all-NLI gives a large boost in performance for Time of Event, Double Negation and Lexical Semantics, but offers relatively little help for Classifiers, Idioms, pro-drop, and several other linguistic categories.

We do want to emphasize that our results also show in several cases, the cross-lingual transfer strategy clearly performs worse than monolingual fine-tuning. For instance, as we see from the results on Chinese HANS 6.5.2, cross-lingual transfer have similar performance with monolingual models for the simpler lexical overlap examples, but falls behind

| gold \ prediction | entailment | neutral | contradiction |
|---|---|---|---|
| entailment | 73 | 9 | 2 |
| neutral | 19 | 56 | 10 |
| contradiction | 13 | 15 | 53 |

Table 6.12: Confusion matrix of XLM-R (En-all-NLI) on the idioms section of diagnostics

(by more than 10% in accuracy) when evaluated on the harder sub-sequence examples. Other such cases include the spelling-error in stress tests and Boolean reasoning in semantic fragments, where we see a cross-lingual transfer lagging behind monolingual fine-tuning. Considering size of the training data (1,212k English data for cross-lingual transfer vs. 50k OCNLI data for monolingual models), this shows that there is still a limit for what can be achieved by cross-lingual transfer when coupled with massive amounts of data that are not in the target language (Chinese in our case).

The most striking result in this chapter is the zero-shot transfer ability of XLM-R in the uniquely Chinese linguistic phenomena. How is it possible that XLM-R can make correct predictions for cases involving Chinese idioms when it is only fine-tuned on English data? In this section we perform an error analysis on the diagnostics with uniquely Chinese linguistic features, and then discuss the possible reasons for XLM-R's surprisingly good performance, and finally point out limitations of the current work.

**Error analysis of the idioms** Now we analyze the errors for zero-shot transfer learning of XLM-R on the idioms section of the diagnostics, which has 250 questions in total. An XLM-R model fine-tuned on all English NLI data achieves an accuracy of 74.3%, as shown in Table 6.9, which is even higher than the best results from a model fine-tuned with Chinese data alone (70.4% for XLM-R and 69.3% for monolingual RoBERTa).

Looking at the confusion matrix in Table 6.12, the zero-shot model is more likely to wrongly label non-entailment examples as an entailment (32 cases), rather than make a mistake on the entailment cases (11 cases). This partly shows that the model still heavily relies on the surface lexical items, since in our design, we use the surface meanings of the

| idx | idiom<br>*gloss*<br>figurative meaning | premise | hypothesis | gold<br>pred |
|---|---|---|---|---|
| 92 | s e R云<br>*steady walk blue cloud*<br>steady and swift promotion to a high position | Ä 6了 • „ s ? 从d s e R云。<br>Huang married the daughter of the governor and from then on "steady walk blue cloud". | Ä ý ( R云上e L。<br>Huang can walk on the blue cloud. | C<br>E |
| 95 | « • ¥ Æ<br>*lay liver drop gallbladder*<br>very loyal and genuine (to friends) | 他们à个¤ Æ t / « • ¥ Æ „ D 。<br>They have known each other for years and are brothers of "expose liver drop gallbladder" | 他们à个Š • Œ Æ ý « ( «上。<br>They lay their livers and gallbladders on themselves. | C<br>E |
| 39 | Ï ™ g N<br>*hidden dragon crouching tiger*<br>inconspicuous place with very talented people | Ù O û „ q Q ß 6 Ï ™ g N Ù么à M Ø人 S ©人ó不 0。<br>This remote village has "hidden dragon crouching tiger". It is such a wonder that they have several such talented people. | Ù Ì „ 0 O Ü à d人ù U ó ê 一些™、N之{ „ ¨ i。<br>This place is remote and hardly anyone travels there. It only has some animals such as dragons and tigers. | C<br>E |

Table 6.13: Examples where the model took the surface meaning and made mistakes, giving an entailment label to contradictions.

idioms to construct the neutral and contradiction cases, as illustrated in Table 6.13. For instance, in question 39, when the hypothesis adheres to the surface meaning of the idiom Ï ™ g N (*hidden dragon crouching tiger*) and states that there are actual dragons and tigers in the village, the model wrongly predicts that it is really the case. Similarly, for question 95, the people mentioned in the premise and hypothesis are not actually "laying" their organs on themselves, while the model believes so based on the surface meaning of the words such as *liver* and *gallbladder*, rather than their figurative meaning.

**Limitations** Our research raises several questions as to why and how this surprising cross-lingual transfer happens.

- Our intuition is that XLM-R is learning about idioms during pre-training and retaining this knowledge when fine-tuned on downstream tasks such as NLI. To see if this is true, one could try to remove or ablate idioms from the model's pre-training and re-train the model from scratch. The pre-training data does not need to include 100

languages, because this will be prohibitively expensive to train (XLM-R is trained on 2.5 TB data). Having English and Chinese pre-training data will possibly be enough, as long as under one setting the Chinese idioms in the pretraining data are removed.

- Using the more recent "data removal" technique (Guo et al., 2020), one can try to remove idioms in the same, yet more cost-effective way. However, as the data removal method so far has been applied to much simpler machine learning models, this will involve considerable work on updating and generalizing the technique to more complex transformer models.

- Evaluate on NLI problems in other languages. One can design NLI problems involving the reasoning of idioms or language-specific linguistic phenomena in other languages, and compare XLM-R's performance with monolingual models in those languages. If zero-shot, cross-lingual transfer of XLM-R works well on most of or even all the languages, we will be more confident in stating that XLM-R has indeed learned language-agnostic representation of meaning.

We believe our new challenge datasets in this chapter will be invaluable resources for work along these lines, and our results can serve as the starting point to opening up the blackbox of multilingual neural models.

## 6.7 Summary

In this chapter, we examined the cross-lingual transfer ability of XLM-R in the context of Chinese NLI, by evaluating XLM-R and monolingual Chinese models, fine-tuned with different types of data, on four sets of adversarial/probing datasets. Our main finding is that zero-shot transfer learning works relatively well in most cases (better than the best monolingual models), and that adding high-quality Chinese data to the English NLI data for fine-tuning usually achieves the best result. However, in most evaluation data, models are still outpaced by human performance, highlighting the room for improvement and the

need for new non-English diagnostics to benchmark cross-lingual transfer and Chinese NLI models. Our datasets and results also show limitations of our work on opening up the blackbox of multilingual transformers and answering questions such as why and how cross-lingual transfer works.

**Contributions** We see several contributions of our work: 1) this is one of the first studies that investigate cross-lingual transfer based on linguistic categorization, which we hope can inspire a line of future research on linguistic-oriented analysis of the multilingual models; 2) we provide a suite of adversarial and probing evaluation datasets in Chinese NLI, which will facilitate NLU research in Chinese; 3) we present empirical results on when zero-shot, cross-lingual transfer works, as well as why we still need language-specific datasets in an era where machine-translated data are too easy to obtain and widely used.

# CHAPTER 7

# CONCLUSION

In this dissertation, we have presented two lines of research on natural language inference. The first line proposes a logic-based, symbolic inference engine for NLI. The second line creates the first large-scale NLI dataset in Chinese, as well as four challenging evaluation datasets, with the aim of examining the reasoning ability of neural models.

## 7.1 Summary of the chapters

Concretely, in chapter 3, we extended the van Benthem algorithm (van Benthem, 1986) for polarity annotation, and designed and implemented a system to perform polarity annotation based on parse trees from CCG parsers. Evaluation on a small expert-crafted corpus of polarized sentences showed that our system `ccg2mono` has a wider coverage than the dependency-based `NatLog` system. Chapter 4 built on the `ccg2mono` system and proposed an inference engine which relied exclusively on monotonicity and natural logic rules. The knowledge base for the inference engine `MonaLog` was built from WordNet and preorders extracted from input premises. `MonaLog` then generates the inferences with a replacement operation, and returns *entailment* if the hypothesis is in the set of generated entailed sentences. Experiments on two NLI corpora — section one of FraCaS and the full SICK — showed that `MonaLog` performed competitively compared to other logic-based systems on FraCaS and the corrected SICK. This suggests that a symbolic system relying on monotonicity and natural logic is capable of solving a substantial part of the two NLI/RTE corpora that we studied.

Chapter 5 and chapter 6 introduced the first large-scale Chinese NLI corpus, OCNLI, and another four challenging Chinese NLI evaluation datasets that targeted different aspects of reasoning. Chapter 5 compared various schemes of eliciting NLI data, and then proposed

an annotation scheme that produced more challenging NLI data than previous annotation procedure would produce. Experiments with several neural models including the most powerful Chinese transformer models suggested that OCNLI is challenging to the models. Chapter 6 started with the creation of four evaluation datasets, consisting of 17 fine-grained categories. These include both expert-written and synthesized data, adversarial attacks and linguistically oriented probes. Our experiments showed that the XLM-R model performed surprisingly well under zero-shot transfer learning. That is, when fine-tuned only on English data, it achieved even higher score than monolingual Chinese models fine-tuned with Chinese data, for 3 out of 4 challenging datasets we constructed. What was especially surprising was that on 3 out of 5 uniquely Chinese linguistic phenomena, the zero-shot transfer learning results of XLM-R were better than those of monolingual Chinese models. Potential reasons for this surprising result were discussed, and it was also pointed out that the scores of even the best neural model were still far behind human scores, suggesting a lot of room of improvement for the reasoning abilities of the neural models.

## 7.2 Contributions of the dissertation

This dissertation makes several contributions to the field of NLI.

First, it provides the community with the first dedicated system `ccg2mono` for monotonicity annotation. `ccg2mono` uses parse trees from wide-coverage CCG parsers, and can perform polarity annotation on tokens as well as constituents.

Second, a new, light-weight inference engine called `MonaLog` is proposed, which follows the natural logic tradition in NLI. `MonaLog` performs competitively compared to previous symbolic systems, and is capable of generating high-quality inferences as augmented training data for machine learning models, a unique feature that bridges symbolic and neural modeling.

Third, this dissertation contributed to NLI resources outside of English by introducing the first large-scale NLI corpus for Chinese, and four probing NLI datasets for Chinese.

This will benefit (and has already benefited) the Chinese NLP community, as well as researchers interested in multilingual NLI research.

Finally, extensive experimental results with neural models on NLI were reported, showing that cross-lingual transfer of the multilingual transformer models perform better than expected. The results also expose the weaknesses of the current Chinese as well as multilingual models, suggesting that there is much room for improvement.

## 7.3 Outlook

Looking to the future, there are several directions of research that can be pursued.

For symbolic models, the most promising one seems to be combining them with neural models to increase coverage. As we discussed in chapter 4.6, symbolic systems are usually too brittle for real world data, which are messy and diverse. Thus a hybrid of symbolic and neural models seems to be a good choice that takes advantage of the both worlds: foundation in linguistic and logic theories of the symbolic models, and wide coverage and robustness of the neural ones. One example of such model is the NeuralLog model (E. Chen et al., 2021), which uses a controller for symbolic inference generation (with the help of Sentence-BERT, Reimers and Gurevych (2019)), and the similarity between chunks of premise and hypothesis—calculated using ALBERT (Lan et al., 2019)—to handle syntactic variation. Of course, finding a reasonable and feasible way to combine these two types of models will be a challenge. In chapter 4 we experimented with a naive method which considers the symbolic model as the default system and falls back to the neural model if the symbolic model cannot find enough evidence/proof for either entailment or contradiction. It is worth experimenting using an ensemble of logic and neural models, for instance.

On the other hand, claims made with respect to neural models, in particular the transformer models, need to be more carefully examined and interpreted. While data-intensive and compute-intensive models such as BERT or RoBERTa have achieved very high scores on English datasets, sometimes even above human score, this should not be taken as an

indication that NLI is too easy or has been solved. As our work in chapter 5 shows, enhanced annotation procedures will produce data that better represents the real challenges in natural language where a gap between the human and best model's performance is still large, regardless of having ever-increasing model size and pre-training data.[1] The poor performance (around 60%) of the transformer models on the more recent adversarially created datasets such as ANLI (Nie et al., 2020a) suggests the same story. That is, there are fundamental reasoning skills that simply cannot be acquired by the current transformer models, or at least under the current model architecture and training strategy.

I can see two directions moving forward for neural modeling. First is to understand what exactly the neural models are doing, either via probing classifiers (Belinkov et al., 2020) or other techniques to understand the weights in the neural networks, for example visualization techniques (Reif et al., 2019), analyzing their geometrical properties (Ethayarajh, 2019), among others. The second is to perform comprehensive analysis of the model errors, for instance by checking on which reasoning categories the models are failing (Williams et al., 2020). NLI can provide much insight into the blackbox of the neural models as it is an ideal probing task to test a model's understanding abilities on specific phenomena.

In sum, this dissertation has presented work on both the symbolic and neural approaches to natural language inference. We hope to see more research on combining these two approaches and on the interpretability of the neural models.

[1]For instance, as of May 2021, the best model performance on OCNLI is 83.67%, still far behind the human score of 90.3%. C.f., https://www.cluebenchmarks.com/nli.html

# APPENDIX A

# EXAMPLE LEXICON WITH SEMANTIC TYPES WITH MARKINGS FOR CCG2MONO

In this section, we list the (most common) semantic categories from the CCG parsers for selected words.

| words | semantic type | notes |
|---|---|---|
| quantifiers | | |
| some, a, an | $N$ Ñ *NP* | |
| every, all, each | $N$ Ñ *NP* | |
| no | $N$ Ñ *NP* | |
| most | $N$ Ñ *NP* | $N\{N$ fixed to $N\{NP$ for both parsers |
| any | $N$ Ñ *NP* or $N$ Ñ *NP* | depending on its function (FCI or NPI) |
| few | $N$ Ñ *NP* , same as *few* | (MacCartney and Christopher D. Manning, 2007) |
| at most *n* | $N$ Ñ *NP* , *n* is upward entailing | |
| at least *n* | $N$ Ñ *NP* , *n* is downward entailing | |
| fewer/less than *n* | same as *at most n* | |
| more than *n* | same as *at least n* | |
| the | $N$ Ñ *NP* | |
| nouns/pronouns | | |
| Fido, John | *NP* | |
| dog, cat | $N$ | |

| | | |
|---|---|---|
| pronouns (NP leaf) | *NP* | she, he, it, etc. |
| relative pronoun | | |
| that, who, which | p*NP* Ñ $S$q Ñ p$N$ Ñ $N$q | subjRC: a dog who barks ... |
| that, who, which | p*NP* Ñ $S$q Ñ p$N$ Ñ $N$q | objRC: a dog who every cat likes ... |
| to | | |
| to | p*NP* Ñ $S$q Ñ p*NP* Ñ $S$q | as in *want to* |
| prepositions | | |
| in | *NP* Ñ $PP$ | he puts it **in** the box |
| in | *NP* Ñ pp*NP* Ñ $S$q Ñ p*NP* Ñ $S$qq | He sleeps **in** France (no assumption on NP markings) |
| in | *NP* Ñ p*NP* Ñ *NP*q | the man **in** France sleeps |
| in | *NP* Ñ p$S$ Ñ $S$q | in theory, ... |
| without | *NP* Ñ pp*NP* Ñ $S$q Ñ p*NP* Ñ $S$qq | |
| adjectives | | |
| good | $N$ Ñ $N$ | $N$z$N$ |
| fake | $N$ Ñ $N$ | privative adjectives |
| verbs | | |
| lack, fail, prohibit, refuse | p*NP* Ñ Sq Ñ p*NP* Ñ Sq | (MacCartney and Christopher D. Manning, 2007) |
| intransitive | *NP* Ñ *S* | walk |
| transitive | *NP* Ñ p*NP* Ñ Sq | devour |
| *do/did* | p*NP* Ñ Sq Ñ p*NP* Ñ Sq | as in *do/did not* |
| *put* with PP | *NP* Ñ p*PP* Ñ p*NP* Ñ Sqq | as in *put it on the table* |
| *ask* with PP | *PP* Ñ p*NP* Ñ Sq | as in *ask about it* |
| modal verbs | | |
| can/should/... | p*NP* Ñ Sq Ñ p*NP* Ñ Sq | p$S$z$NP$q{p$S$z$NP$q |

| adverbs | | |
|---|---|---|
| fast | p*NP* Ñ $S$q Ñ p*NP* Ñ $S$q | |
| not, n't | p*NP* Ñ $S$q Ñ p*NP* Ñ $S$q | p$S$z$NP$qzp$S$z$NP$q |
| connectives | | |
| if | $S$ Ñ p$S$ Ñ $S$q | p$S${$S$q{$S$ |
| then | $S$ Ñ $S$ | CandC: $S${$S$. easyccg: $N${$N$ (wrong) |

Table A.1: Example lexicon with markings on the types

# APPENDIX B

# APPENDICES FOR OCNLI

## B.1 Instructions for Hypothesis Generation

(*the instructions are originally in Chinese; translated to English for this paper*)

Welcome to our sentence writing experiment. Our aim is to collect data for making inferences in Chinese. In this experiment, you will see a sentence (A), which describes an event or a scenario, for example:

> Sentence A:
>
> **John won the first prize in his company's swimming competition last year.**

You task is to write three types of sentences based on the information in sentence A, as well as your common sense.

- Type 1: a sentence that is definitely true, based on the information in sentence A, e.g.,
  - John can swim
  - John won a prize last year
  - John's company held a swimming competition last year
- Type 2: a sentence that might be true (but might also be false), based on the information in sentence A, e.g.,
  - John's company held the swimming competition last March
  - Tom ranked second in last year's swimming competition
  - John can do the butterfly style

- Type 3: a sentence that cannot be true, based on the information in sentence A, e.g.,
  - John has not swum before
  - John did not get any prize from the company's swimming competition last year
  - John's company only hold table tennis competitions

You will see 50 sentences A. For each sentence A, you need to write three sentences, one for each type. In total you will write 150 sentences. If there is a problem with sentence A, please mark it as "x". Please refer to FAQ for more examples and further details of the task.

### B.2 Model Details and Hyper-parameters

We experimented with the following models:

- Continuous bag-of-words (CBOW), where each sentence is represented as the sum of the embeddings of the Chinese characters composing the sentence, which are then passed on to a 3-layer MLP.
- Bi-directional LSTM (biLSTM), where the sentences are represented as the average of the states of a bidirectional LSTM.
- Enhanced Sequential Inference Model (ESIM), which is MNLI's implementation of the ESIM model (Q. Chen et al., 2017).
- BERT base for Chinese (BERT), which is a 12-layer transformer model with a hidden size of 768, pre-trained with 0.4 billion tokens of the Chinese Wikipedia dump (Devlin et al., 2019). We use the implementation from the CLUE benchmark (Xu et al., 2020)[1].

[1] https://www.cluebenchmarks.com/

- RoBERTa large pre-trained with whole word masking (wwm) and extended (ext) data (RoBERTa), which is based on RoBERTa (Liu et al., 2019) and has 24 layers with a hidden size of 1024, pre-trained with 5.4 billion tokens, released in (Cui et al., 2019). We use the implementation from the CLUE benchmark.

For CBOW, biLSTM and ESIM, we use Chinese character embeddings from S. Li et al. (2018)[2], and modify the implementation from MNLI[3] to work with Chinese.

Our BERT and RoBERTa models are both fine-tuned with 3 epochs, a learning rate of 2e-5, and a batch size of 32. Our hyper-parameters deviate slightly from those used in CLUE and (Cui et al., 2019)[4], because we found them to be better when tuned against our dev sets (as opposed to XNLI or the machine translated CMNLI in CLUE).

### B.3 More Examples from OCNLI

We present more OCNLI pairs in Table B.1.

[2] https://github.com/Embedding/Chinese-Word-Vectors
[3] https://github.com/NYU-MLL/multiNLI
[4] https://github.com/ymcui/Chinese-BERT-wwm/

| Premise | Genre<br>Level | Majority label<br>All labels | Hypothesis |
|---|---|---|---|
| / ` 他²Ù个^ 意<br>Yes, look, what he talked about is very interesting. | TV<br>hard | **Entailment**<br>E E N E E | 他²„Ù个 w了 „sè<br>What he talked about has caught my attention. |
| •9»Ö œ åå D î ~ “紧6š 专èL?ÕÄ nÝ付ú›³ŒW4„ œ å öÿ0” „¥l<br>(We need to) solve the problem of delaying wages for the migrant workers at its root and act promptly to lay out specific administrative regulations to ensure those hardworking migrant workers receive the wages they deserve in a timely manner. | GOV<br>easy | **Neutral**<br>N E N N N | 专èL?ÕÄ/ã³Ö åDî ~„9, „<br>(Designing) specific administrative regulations is the most fundamental way of solving the issue of wage delays. |
| `•Þef,10( Ù ?.<br>If you are back, you can stay in my old house. | PHONE<br>hard | **Contradiction**<br>C C C C C | ¡?<br>I don't have a house. |
| A •Þ»,A一 份•Þe.<br>Going there at the end of October, be back at the end of November. | PHONE<br>medium | **Contradiction**<br>C C C C C | •(£¹F两个 MÞe。<br>Will stay there for two months before coming back. |
| C,ù, ', ,Ù.<br>Er, yes, I may have (it), here. | PHONE<br>hard | Neutral<br>N N N N N | /+人óî Ù个东•<br>Someone else is trying to borrow this from me. |
| e一v一v0从9上Ç» }ÏÛ了一 G一G„è<br>Bridge after bridge was passed above the boat, just like going through door after door. | LIT<br>medium | **Entailment**<br>E E E E E | 不b一§e<br>There is more than one bridge. |
| dô ºLnM¤为 Ù!ê Z( ®b®-Ï à' î~(于Ï<br>It is generally believed by the media that the Liberal Democratic Party are going to lose their seats. The problem is how many. | NEWS<br>medium | **Contradiction**<br>C C C N N | ºLnM¤为 Ù!ê Z• «q ú ®b。<br>It is generally believed by the media that the Liberal Democratic Party will be ousted from the House of Representatives. |

Table B.1: More examples from OCNLI.

# APPENDIX C

# APPENDICES FOR CHINESE NLI PROBING DATASETS

## C.1 Details about dataset creation and examples

In this section, we list example NLI pairs and their translations. For examples of the Chinese HANS and stress tests, see Table 6.3. For the expanded diagnostics, see Table 6.4. For the semantic/logic probing dataset, see Table 6.5.

**Antonym** Instances with simple antonym substitution have been used in previous studies (Glockner et al., 2018; Naik et al., 2018) for creating stress tests. That is, we replace a word in the premise with its antonym, producing a contradictory hypothesis. For example, using *John loves Mary* as the premise, after replacing the verb with its antonym, we get a contradiction hypothesis: *John hates Mary*. After looking at the initially generated data, we decided to replace only the nouns and adjectives with their antonyms since such replacements are most likely to result in contradictory hypotheses that are grammatical.[1]

**Synonym** Similar to the antonym setting, replacing a word in the premise with its synonym has also been used in the literature to generate adversarial examples (Glockner et al., 2018). We follow Glockner et al. (2018) to consider such examples as having an entailment relation. After inspecting the initially generated data, we decided to perform replacements only to verbs and adjectives. To ensure the quality of synonyms, we rank the synonyms from a commonly used synonym dictionary by their vector similarity to the original word, and pick the top ranking synonym.[2]

[1] We use the LTP toolkit (https://github.com/HIT-SCIR/ltp) to annotate the POS tags and the antonym list from https://github.com/liuhuanyong/ChineseAntiword.

[2] We use the synonym list from https://github.com/Keson96/SynoCN and the similarity score from the Python package Synonymshttps://github.com/chatopera/Synonyms.

| Heuristic | entailment | contradiction | neutral |
|---|---|---|---|
| lexical overlap | 441 | 647 | 340 |
| subsequence | 100 | 193 | 220 |
| Total | 541 | 840 | 560 |

Table C.1: Distributional statistics of the synthesized Chinese HANS.

**Distraction** We created the distraction data basically following the stress test setting (Naik et al., 2018) but sorted the data by the position of distraction added. Tautologies/true statements are added to either the premise or hypothesis part, in the meantime, we investigate the influence of negation words in such statements as well:

- **Premise-no-negation**: A tautology/true statement is added to the end of the premise in order to examine the ability of the model to capture information from a large context.
- **Premise-negation**: A tautology/true statement described with negation word is added to the premise.
- **Hypothesis-no-negation**: Instead of inserting to the premise, a tautology/true statement is added to the hypothesis this time. This test allows us to investigate the ability of the model to reason about the hypothesis with non-related information.
- **Hypothesis-negation**: We insert a tautology/true statement described with negation word to the hypothesis.

Only two tautologies are used in the original paper in the experiment setting. In this paper, to thoroughly examine the influence of different true statements, we designed 50 tautology/statements varied in three factors: length, out-of-vocabulary, and negation word. There are 25 statements pairs in total (1 tautology and 24 true statements), each statements pair includes a true statement and its corresponding true statement with negation form. All the statements range from 5 to 16 characters. For the true statements in negation form, two

common Chinese negation words 不 and ¡ are used to represent the negation meaning. For the 24 true statements pairs, half of them contains at least one Out-of-Vocabulary word in OCNLI.

Experiment results indicate that the length, Out-of-Vocabulary word, and choice of negation word will not affect the results, so we do not report the results of these variables.

**Spelling** Inspired by Naik et al. (2018), we generate a set of data containing "spelling errors" by replacing one random character in the hypotheses with its homonym, which is defined as a character with the same *pinyin* pronunciation ignoring tones. We also limit the frequency of the homonym as within the range of 100 to 6000 so that the character is neither too rare nor too frequent.

**Numerical reasoning** Inspired by Naik et al. (2018), we created a probing test for numerical reasoning. We extracted sentences from Ape210k (Zhao et al., 2020), a large-scale math word problem dataset containing 210K Chinese elementary school-level problems[3]. We generate entailed, contradictory and neutral hypotheses for each premises, following the heuristic rules below (adapted from Naik et al. (2018)):

1. **Entailment:** Randomly choose a number x and change it to y from the hypothesis. If the y ¡ x, prefix it with one phrase that translate to "less than"; if y x, prefix it with one phrase that translate to "more than".

   Premise: *Mary types 110 words per minute.* Hypothesis: *Mary types less than 510 words per minute.*
2. **Contradiction:** Perform either 1) randomly choose a number x from the hypothesis and change it; 2) randomly choose a number from the hypothesis and prefix it with one phrase that translate to "less than" or "more than".

[3]We split all problems into individual sentences and filter out sentences without numbers. Then we remove sentences without any named entities ("PERSON", "LOCATION" and "ORGANIZATION") using tools provided by LTP toolkit (Che et al., 2020).

Premise: *Mary types 110 words per minute.* Hypothesis: *Mary types 710 words per minute.*

3. **Neutral:** Reverse the corresponding entailed premise-hypothesis pairs.

   Premise: *Mary types less than 510 words per minute.* Hypothesis: *Mary types 110 words per minute.*

The result contains 2,871 unique premise sentences and 8,613 NLI pairs.

Note that for the **Distraction**, **Antonym**, **Synonym** and **Spelling** subsets, we use equal number of seed sentences from OCNLI_dev and XNLI_dev so as not to bias any of the two existing Chinese NLI datasets.

### C.2 Templates for Chinese HANS

The templates for Chinese HANS are presented in Table C.2 and Table C.3.

### C.3 Details about hyperparameters

Please see Table C.4 for details about the hyperparameters of different models. The learning-rate search space for RoBERTa is 1e-5, 2e-5, 3e-5, 4e-5 and 5e-5, for XLM-R it is 5e-6, 7e-6, 9e-6, 2e-5 and 5e-5.

### C.4 Results for Chinese-to-English transfer

We present Chinese-to-English transfer results for the English stress tests in Table C.6, for English HANS in Table C.5, for English diagnostics in Table C.8 and Table C.9, for English semantic probing data in Table C.7. As mentioned in the main text, for most of the cases transfer learning does not work well mostly likely due to the small size of OCNLI.

Specifically, for English HANS, XLM-R fine-tuned with OCNLI is about 13 percentage points below the best English monolingual model. For stress tests, the gap is about 5 percent. For semantic probing data, XLM-R with OCNLI performs better than monolingual

model fine-tuned with MNLI, but is 5 percent behind monolingual model fine-tuned with all English NLI. For the English diagnostics, XLM-R with OCNLI is 7 percent behind RoBERTa fine-tuned with MNLI.

| Category | Template (Premise Ñ Hypothesis) | Example |
|---|---|---|
| Entailment:<br>« sentence | $N_1$ « V （ $N_{loc}$ 了。<br>Ñ $N_1$ （ $N_{loc}$ 。 | z / ¶ « s ( ) ‡ † 了。<br>The artist is locked in the planetarium.<br>Ñ z / ¶ ( ) ‡ † 。<br>The artist is inside the planetarium. |
| Entailment:<br>PP-drop | $N_1$ （ $N_{loc}$ LC V $N_2$ 。<br>Ñ $N_1$ V $N_2$ 。 | † ü ( – a † D Ñ • d R 。<br>The leader is drinking beer near the coffee shop.<br>Ñ † ü • d R 。<br>The leader is drinking beer. |
| Entailment:<br>Adverb-Þ | Þ V $N_1$ „ $N_2$ ý É – ADJ 。<br>Ñ $N_2$ V $N_1$ 。 | Þ Ý g „ v ý É – í ù 。<br>Even graduate students watching a drama feel excited.<br>Ñ v Ý g 。<br>The graduate students are watching a drama. |
| Entailment:<br>Choice | P$N$ 不 / $N_1$, F / / $N_2$ 。<br>Ñ P$N$ / $N_2$ 。 | y 不 / ; ,F / / Ñ f ¶ 。<br>She is not a doctor, but a scientist.<br>Ñ y / Ñ f ¶ 。 She is a scientist. |
| Entailment:<br>Adverb-drop | $N_1$ ADV V Ç $N_2$ 。<br>Ñ $N_1$ V Ç $N_2$ 了。 | Õ ˜ œ 6 ² Ç Ý 。 As expected, the judge has made jokes.<br>Ñ Õ ˜ ² Ç Ý 。 The judge has made jokes. |
| Contradiction:<br>Negation | $N_1$ ¡ V Ç $N_2$ 。<br>Û $N_1$ V Ç $N_2$ 。 | ; ¡ Ç 5 q 。 The doctors has never watched movies.<br>Û ; Ç 5 q 。 The doctor has watched movies. |
| Contradiction:<br>Double Negation | $N_1$ 不 / 不V $N_2$ 。<br>Û $N_1$ 不V $N_2$ 。 | å 不 / 不 H m 。<br>It's not the case that cleaners do not eat lunch.<br>Û å 不 H m 。 Cleaners do not eat lunch. |
| Contradiction:<br>Swap | P$N$ Š $N_1$ V （ $N_{loc}$ 了。<br>Û $N_1$ ŠP$N$ V （ $N_{loc}$ 了。 | 们 Š ö L L X Y ( 5 q b 了。<br>We left the bank clerk in the cinema.<br>Û ö L L X Š 们 Y ( 5 q b 了。<br>The bank clerk left us in the cinema. |
| Contradiction:<br>Choice | $N_1$ , e ó $V_1$ $N_2$, Ó œ $V_2$ $N_3$ 了。<br>Û $N_1$ , e ó $V_2$ $N_3$ 。 | Y ˆ , e ó • d R , Ó œ • Ü 了。<br>The professor was thinking to drink beer but ate watermelon instead.<br>Ù Y ˆ , e ó • Ü 。<br>The professor was thinking to eat watermelon. |
| Contradiction:<br>Condition | $N_1$ A$dv_{cnd}$ $V_1$ Ç $N_2$ 1 } 了。<br>Û $N_1$ $V_1$ Ç $N_2$ 。 | ¹ ¹ , œ » Ç ™ ä 1 } 了。<br>If only the younger sister had gone to Mongolia.<br>Û ¹ ¹ » Ç ™ ä 。<br>The younger sister has gone to Mongolia. |
| Neutral:<br>Choice | $N_1$ Œ $N_2$ P$N$ $V_1$ v 中一个。<br>Û P$N$ $V_1$ $N_1$ 。 / P$N$ $V_1$ $N_2$ 。 | Y ˆ Œ Ï ,他 œ " v 中一个。<br>He likes either the professor or the manager.<br>Ù 他 œ " Ï 。 He likes the manager. |
| Neutral:<br>Argument Drop | $N_1$ „ $N_2$ （ $V_1$ $N_3$ 。<br>Û $N_1$ （ $V_1$ $N_3$ 。 | Ø 书 „ （ ó 。<br>The secretary's younger brother is dancing.<br>Ù Ø 书 ( ó 。 The secretary is dancing. |
| Neutral:<br>Drop • | Ï 个 $N_1$ ý • $V_1$ $N_2$ 。<br>Û Ï 个 $N_1$ ý $V_1$ $N_2$ 。 | Ï 个 å ý • 买 • Ü 。<br>Every cleaner wants to buy watermelon.<br>Ù Ï 个 å ý 买 • Ü 。<br>Every cleaner is going to buy watermelon. |
| Neutral:<br>Adverb Drop | $N_1$ Adv $V_1$ Ç $N_2$ 。<br>Û $N_1$ $V_1$ Ç $N_2$ 。 | å < 乎 Ç é m 。<br>The cleaner seems to have eaten breakfast.<br>Ù å Ç é m 。 The cleaner has eaten breakfast. |
| Neutral:<br>Adverb Drop | ¡ Õ Á $N_1$ $V_1$ Ç $N_2$ 。<br>Û $N_1$ $V_1$ Ç $N_2$ 。 | ¡ Õ Á 7 7 V Ç • ¢ ÿ 。<br>It cannot be proven that the grandfather has sold tomatoes.<br>Ù 7 7 V Ç • ¢ ÿ 。<br>The grandfather has sold tomatoes. |

Table C.2: Template Examples of Lexical Overlap Heuristic in Chinese HANS.

| Category | Template | Example |
|---|---|---|
| Entailment:<br>Adverb Drop | Adv $N_1$ $V_1$ $N_2$ 了。<br>Ñ $N_1$ $V_1$ $N_2$ 了。 | Í c 们 橘 P 了。<br>Anyhow, we ate tangerines.<br>Ñ 们 橘 P 了。<br>We ate tangerines. |
| Entailment:<br>Adverb Drop | Adv $N_1$ $V_1$ Ç $N_2$。<br>Ñ $N_1$ $V_1$ Ç $N_2$。 | œ 6 å , Ç ó 乐。<br>As expected, the cleaner has listened to music.<br>Ñ å , Ç ó 乐。<br>The cleaner has listened to music. |
| Contradiction:<br>Drop 以为 | $N_1$ 以为 $N_2$ $V_1$ $N_3$ 了。<br>Û $N_2$ $V_1$ $N_3$ 了。 | Ñ f ¶ 以为 Õ ˜ ó 了。<br>The scientist thought that the judge danced.<br>Ñ Õ ˜ ó 了。<br>The judge danced. |
| Contradiction:<br>Drop | ô $N_1$ ý / $V_1$ $N_2$ „。<br>Û $N_1$ ý / $V_1$ $N_2$ „。 | ô Ï ý / S † & „。<br>Who told you that managers all wear ties?<br>Ñ Ï ý / S † & „。<br>Managers all wear ties. |
| Contradiction:<br>Drop | Num t ž ° Ï 个 $N_1$ ý $N_2$。<br>Û Ï 个 $N_1$ ý $N_2$。 | 三 t ž ° Ï 个 ¿ ý 京 g â。<br>In three years, the goal will be realized that<br>every county has a Chinese operator troupe.<br>Ñ Ï 个 ¿ ý 京 g â。<br>Every county has a Chinese operator troupe. |
| Neutral:<br>Adv, RC | Adv $N_1$ $V_1$ „ $N_2$ $V_2$ Ç $N_3$。<br>Û $N_2$ $V_2$ Ç $N_3$。 | ï ý Ø 书 œ " „ z / ¶ 买 Ç È Æ Ü。<br>Maybe the artist that the secretary likes<br>has bought Hami melon.<br>Ñ z / ¶ 买 Ç È Æ Ü。<br>The artist has bought Hami melon. |
| Neutral:<br>Drop | / 不 / $N_1$ $V_1$ „ $N_2$。<br>Û / $N_1$ $V_1$ „ $N_2$。 | / 不 / † ü œ " „ v 。<br>Let's see if (he/she) is the kind of students the leader likes.<br>Ñ / † ü œ " „ v 。<br>(He/she) is the kind of students the leader likes. |
| Neutral:<br>Drop Adverb | Adv $N_1$ $V_1$ $N_2$ 了。<br>Û $N_1$ $V_1$ $N_2$ 了。 | 也 ¸ Ï , L g 了。<br>Maybe the manager listened to the operator.<br>Ñ Ï , L g 了。<br>The manager listened to the operator. |
| Neutral:<br>Þ | Þ $V_1$ $N_1$ „ $N_2$ ý É – ADJ。<br>Û $N_2$ ý É – ADJ。 | Þ , ò „ å ý É – é。<br>Even the cleaners who listen to the Kun opera<br>thinks it's too early.<br>Ñ å ý É – é。<br>Even cleaners think it's too early. |

Table C.3: Template Examples of Sub-sequence Heuristic in Chinese HANS

| Model | Traning Data | Max Length | Epoch | Learning Rate |
|---|---|---|---|---|
| RoBERTa | zh MT: XNLI-small | 128 | 3 | 3.00E-05 |
| RoBERTa | zh MT: XNLI | 128 | 3 | 2.00E-05 |
| RoBERTa | zh ori: OCNLI | 128 | 3 | 2.00E-05 |
| RoBERTa | zh: OCNLI + XNLI | 128 | 3 | 3.00E-05 |
| XLM-R | zh ori: OCNLI | 128 | 3 | 5.00E-06 |
| XLM-R | zh MT: XNLI | 128 | 3 | 7.00E-06 |
| XLM-R | zh: OCNLI + XNLI | 128 | 3 | 7.00E-06 |
| XLM-R | en:MNLI | 128 | 3 | 5.00E-06 |
| XLM-R | en: En all NLI | 128 | 3 | 7.00E-06 |
| XLM-R | mix: OCNLI + En all NLI | 128 | 3 | 7.00E-06 |
| XLM-R | mix: XNLI + En all NLI | 128 | 3 | 7.00E-06 |

Table C.4: Hyper-parameters used for fine-tuning the models.

| Model | Fine-tuned on | Overall | Lexical_overlap | Subsequence | Constituent | Entailment | Non-entailment |
|---|---|---|---|---|---|---|---|
| RoBERTa | en: En-all-NLI | 76.54 | 96.79 | 67.77 | 65.06 | 99.81 | 53.27 |
| RoBERTa | en: MNLI | 77.63 | 95.60 | **68.08** | 69.21 | 99.74 | 55.52 |
| XLM-R | en: En-all-NLI | 75.72 | 95.52 | 62.99 | 68.63 | 99.91 | 51.52 |
| XLM-R | en: MNLI | 74.80 | 92.92 | 65.24 | 66.23 | 98.83 | 50.76 |
| XLM-R | zh ori: OCNLI | 64.37 | 71.28 | 54.42 | 67.41 | 98.39 | 30.35 |
| XLM-R | zh MT: XNLI | 68.83 | 81.67 | 62.07 | 62.74 | 99.13 | 38.53 |
| XLM-R | zh mix: OCNLI+XNLI | 71.30 | 82.52 | 61.72 | 69.66 | 99.08 | 43.52 |
| XLM-R | mix: OCNLI+En-all-NLI | **78.56** | **96.92** | 64.91 | **73.84** | 99.92 | **57.20** |
| XLM-R | mix: XNLI+En-all-NLI | 74.65 | 93.93 | 60.97 | 69.04 | **99.96** | 49.34 |

Table C.5: Results of English HANS (McCoy et al., 2019).

| Model | Fine-tuned on | Overall | Antonym | Content word swap | Function word swap | Keyboard | Swap | Length mismatch | Negation | Numerical reasoning | Word overlap |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | en: En-all-NLI | 79.48 | 82.91 | **86.22** | 88.71 | **87.8** | **87.48** | **88.28** | 60.25 | 79.26 | 62.85 |
| RoBERTa | en: MNLI | 77.9 | 69.03 | 85.74 | **88.75** | 87.39 | 87.05 | 88.23 | 59.19 | 65.46 | 61.48 |
| XLM-R | en: En-all-NLI | 79.6 | 86.25 | 85.26 | 87.38 | 86.31 | 86.72 | 87.25 | 61.06 | **82.84** | 65.79 |
| XLM-R | en: MNLI | 77.6 | 74.65 | 85.09 | 87.33 | 86.08 | 86.42 | 86.96 | 60.95 | 54.66 | 65.13 |
| XLM-R | zh ori: OCNLI | 74.31 | 72.52 | 75.12 | 77.71 | 76.27 | 76.39 | 77.23 | **72.86** | 55.85 | **72.79** |
| XLM-R | zh MT: XNLI | 77.78 | 65.12 | 85.11 | 86.64 | 85.79 | 85.71 | 85.91 | 63.52 | 43.95 | 71.63 |
| XLM-R | zh mix: OCNLI+XNLI | 77.83 | 66.83 | 84.96 | 86.69 | 85.81 | 85.87 | 85.98 | 63.97 | 51.56 | 68.38 |
| XLM-R | mix: OCNLI+En-all-NLI | **80.01** | **86.33** | 85.22 | 87.4 | 86.26 | 86.77 | 87.23 | 62.52 | 81.79 | 67.54 |
| XLM-R | mix: XNLI+En-all-NLI | 79.38 | 85.27 | 85.35 | 87.2 | 86.28 | 86.74 | 87.22 | 60.29 | 80.5 | 66.19 |

Table C.6: Results of English stress test (Naik et al., 2018).

| Model | Fine-tuned on | Overall | Boolean | Comparative | Conditional | Counting | Monotonicity hard | Monotonicity simple | Negation | Quantifier |
|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | en: MNLI | 51.31 | 43.58 | 39.6 | 66.24 | 63.34 | 61.28 | 60.1 | 37.26 | 39.08 |
| RoBERTa | en: En-all-NLI | 58.72 | 60.18 | 40.28 | **66.3** | 66.22 | 59.6 | 58.98 | 64.46 | **53.74** |
| XLM-R | en: MNLI | 53.54 | 59.16 | 41.62 | 66.3 | 61.72 | 63.26 | **62.82** | 33.52 | 39.92 |
| XLM-R | en: En-all-NLI | 59.85 | **71.58** | 45.18 | 66.3 | 60.4 | 63.86 | 62.02 | 65.68 | 43.78 |
| XLM-R | zh ori: OCNLI | 53.61 | 66.02 | **60.62** | 41.1 | 58.0 | 47.86 | 49.88 | 51.88 | 53.5 |
| XLM-R | zh MT: XNLI | 52.29 | 43.24 | 39.0 | 66.22 | 65.66 | 58.08 | 62.74 | 34.12 | 49.24 |
| XLM-R | zh mix: OCNLI+XNLI | 54.68 | 54.64 | 38.84 | 66.28 | **67.38** | 58.18 | 61.38 | 41.88 | 48.82 |
| XLM-R | mix: OCNLI+En-all-NLI | **60.2** | 71.2 | 42.58 | 66.3 | 62.4 | **64.72** | 60.9 | **68.58** | 44.88 |
| XLM-R | mix: XNLI+En-all-NLI | 60.06 | 65.8 | 46.86 | 66.3 | 65.54 | 61.56 | 61.44 | 68.5 | 44.48 |

Table C.7: Results of English semantic probing datasets (Richardson et al., 2020).

| Model | Fine-tuned on | Overall | active passive | anaphora coreference | common sense | conditionals | anaphora coreference | coordination scope | core args | datives | disjunction | double negation | downward mono-tone | ellipsis implicits | existential | factivity | genitives partitives | intersectivity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | en: MNLI | 66.87 | **62.35** | 67.59 | **69.47** | 62.5 | 78.0 | 63.5 | 69.62 | **85.0** | 39.47 | **92.86** | **19.33** | 65.29 | 65.0 | 62.06 | **95.0** | 60.43 |
| RoBERTa | en: En-all-NLI | **68.03** | 61.76 | **70.0** | 69.33 | **63.75** | **82.5** | **68.0** | **75.77** | 85.0 | **41.58** | 92.14 | 18.67 | **67.65** | 65.0 | **62.35** | 94.0 | 59.57 |
| XLM-R | en: MNLI | 63.03 | 61.76 | 62.76 | 59.73 | 55.62 | 76.0 | 61.5 | 61.54 | 85.0 | 26.84 | 91.43 | 16.0 | 64.12 | 69.0 | 51.47 | 90.0 | 60.0 |
| XLM-R | en: En-all-NLI | 64.57 | 61.76 | 65.17 | 61.47 | 60.0 | 76.0 | 66.0 | 65.77 | 85.0 | 33.16 | 89.29 | 14.0 | 62.94 | **71.0** | 58.53 | 90.0 | **60.87** |
| XLM-R | zh ori: OCNLI | 59.67 | 60.0 | 59.31 | 57.2 | 58.12 | 70.0 | 56.5 | 61.54 | 85.0 | 30.0 | 67.14 | 17.33 | 54.71 | 66.0 | 46.18 | 90.0 | 59.57 |
| XLM-R | zh MT: XNLI | 61.76 | 61.18 | 64.14 | 60.67 | 58.75 | 72.5 | 60.0 | 60.77 | 85.0 | 33.16 | 91.43 | 12.67 | 58.24 | 64.0 | 48.24 | 90.0 | 57.83 |
| XLM-R | zh mix: OCNLI+XNLI | 61.78 | 61.76 | 62.76 | 56.93 | 57.5 | 74.5 | 61.0 | 61.54 | 85.0 | 31.05 | 90.0 | 12.0 | 57.65 | 65.0 | 48.53 | 90.0 | 57.39 |
| XLM-R | mix: OCNLI+En-all-NLI | 64.51 | 61.76 | 63.45 | 61.6 | 58.75 | 76.0 | 66.0 | 67.31 | 85.0 | 35.26 | 90.71 | 15.33 | 60.59 | 68.0 | 60.59 | 91.0 | 60.87 |
| XLM-R | mix: XNLI+En-all-NLI | 64.37 | 61.18 | 64.83 | 62.27 | 61.88 | 73.0 | 65.0 | 65.38 | 85.0 | 35.26 | 91.43 | 14.67 | 63.53 | 70.0 | 57.94 | 94.0 | 60.87 |

Table C.8: Results of English Diagnostics from GLUE-Part I (A. Wang et al., 2018).

| Model | Fine-tuned on | intervals numbers | lexical entailment | morphological negation | named entities | negation | nominalization | non-monotone | prepositional phrases | quantifiers | redundancy | relative clauses | restrictivity | symmetry collectivity | temporal | universal | upward monotone | world knowledge |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | en: MNLI | 54.74 | 66.71 | **89.23** | 57.22 | 66.59 | 82.14 | 56.0 | 86.18 | 78.46 | 79.23 | 63.75 | 55.38 | 67.86 | 56.25 | **84.44** | 76.47 | 48.51 |
| RoBERTa | en: En-all-NLI | **63.16** | **71.57** | 89.23 | 61.11 | **69.02** | 84.29 | **57.33** | 84.41 | 74.62 | 73.85 | 63.75 | 53.08 | **70.0** | **69.38** | 83.33 | 73.53 | **49.55** |
| XLM-R | en: MNLI | 45.79 | 65.57 | 84.62 | 61.11 | 61.95 | 82.14 | 52.0 | 85.88 | **82.69** | 78.46 | 62.5 | 52.31 | 57.14 | 51.25 | 80.0 | 74.12 | 44.03 |
| XLM-R | en: En-all-NLI | 45.79 | 69.71 | 84.62 | 61.67 | 64.15 | **85.71** | 48.67 | 84.71 | 79.23 | 69.23 | 63.12 | 46.15 | 59.29 | 60.0 | 77.78 | 75.29 | 45.82 |
| XLM-R | zh ori: OCNLI | 39.47 | 56.29 | 75.38 | 41.11 | 53.41 | 73.57 | 51.33 | 85.59 | 63.08 | 83.08 | 62.5 | **70.77** | 64.29 | 54.38 | 62.22 | **78.82** | 47.31 |
| XLM-R | zh MT: XNLI | 42.11 | 60.14 | 84.62 | 61.11 | 61.95 | 74.29 | 53.33 | **86.76** | 73.46 | 84.62 | 60.62 | 60.0 | 57.86 | 40.62 | 84.44 | 68.24 | 47.31 |
| XLM-R | zh mix: OCNLI+XNLI | 43.68 | 59.29 | 83.08 | **63.33** | 62.2 | 74.29 | 52.0 | 86.76 | 76.92 | **85.38** | 62.5 | 60.77 | 59.29 | 43.75 | 82.22 | 67.65 | 47.61 |
| XLM-R | mix: OCNLI+En-all-NLI | 45.26 | 69.86 | 85.38 | 62.22 | 65.12 | 85.71 | 50.67 | 85.0 | 74.62 | 69.23 | **68.12** | 47.69 | 60.71 | 56.25 | 77.78 | 75.29 | 45.67 |
| XLM-R | mix: XNLI+En-all-NLI | 44.21 | 67.29 | 86.15 | 62.22 | 63.9 | 83.57 | 49.33 | 84.71 | 75.0 | 70.77 | 66.88 | 46.92 | 58.57 | 57.5 | 74.44 | 74.12 | 45.82 |

Table C.9: Results of English Diagnostics from GLUE-Part II (A. Wang et al., 2018).

# BIBLIOGRAPHY


Abzianidze, Lasha (2014). "Towards a Wide-Coverage Tableau Method for Natural Logic". In: *New Frontiers in Artificial Intelligence - JSAI-isAI 2014 Workshops, LENLS, JURISIN, and GABA, Kanagawa, Japan, October 27-28, 2014, Revised Selected Papers*, pp. 66–82. DOI: 10.1007/978-3-662-48119-6_6.

— (2015). "A Tableau Prover for Natural Logic and Language". In: *Proceedings of EMNLP*, pp. 2492–2502. URL: https://www.aclweb.org/anthology/D15-1296/.

— (2016a). "A natural proof system for natural language". PhD thesis. Tilburg University.

— (2016b). "Natural solution to fracas entailment problems". In: *Proceedings of the *SEM*, pp. 64–74.

— (2017). "LangPro: Natural Language Theorem Prover". In: *CoRR* abs/1708.09417.

Agić, Željko and Natalie Schluter (2017). "Baselines and test data for cross-lingual inference". In: *Proceedings of LREC*. URL: https://arxiv.org/abs/1704.05347.

Alabbas, Maytham (2013). "A dataset for Arabic textual entailment". In: *Proceedings of the Student Research Workshop associated with RANLP 2013*, pp. 7–13.

Amirkhani, Hossein, Mohammad Azari Jafari, Azadeh Amirak, Zohreh Pourjafari, Soroush Faridan Jahromi, and Zeinab Kouhkan (2020). "FarsTail: A Persian Natural Language Inference Dataset". In: *arXiv preprint arXiv:2009.08820*. URL: https://arxiv.org/abs/2009.08820.

Angeli, Gabor and Christopher Manning (2014). "NaturalLI: Natural Logic Inference for Common Sense Reasoning". In: *Proceedings of EMNLP*, pp. 534–545.

Angeli, Gabor, Neha Nayak, and Christopher Manning (2016). "Combining Natural Logic and Shallow Reasoning for Question Answering". In: *Proceedings of ACL*, pp. 442–452.

Angeli, Gabor, Melvin Jose Johnson Premkumar, and Christopher D Manning (2015). "Leveraging linguistic structure for open domain information extraction". In: *Proceedings of ACL*, pp. 344–354.

Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama (2020). "On the Cross-lingual Transferability of Monolingual Representations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4623–4637. DOI: 10.18653/v1/2020.acl-main.421. URL: https://www.aclweb.org/anthology/2020.acl-main.421.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *Proceedings of ICML*.

Basile, Pierpaolo, Marco de Gemmis, Anna Lisa Gentile, Pasquale Lops, and Giovanni Semeraro (2007). "UNIBA: JIGSAW algorithm for Word Sense Disambiguation". In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, pp. 398–401. URL: http://www.aclweb.org/anthology/S/S07/S07-1088.

Belinkov, Yonatan, Sebastian Gehrmann, and Ellie Pavlick (2020). "Interpretability and Analysis in Neural NLP". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 1–5. DOI: 10.18653/v1/2020.acl-tutorials.1. URL: https://www.aclweb.org/anthology/2020.acl-tutorials.1.

Beltagy, Islam, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J Mooney (2016). "Representing meaning with a combination of logical and distributional models". In: *Computational Linguistics* 42.4, pp. 763–808.

van Benthem, Johan (1986). *Essays in Logical Semantics*. Vol. 29. Studies in Linguistics and Philosophy. Dordrecht: D. Reidel Publishing Co., pp. xii+225.

van Benthem, Johan (2008). “A Brief History of Natural Logic”. In: *Logic, Navya-Nyaya and Applications, Homage to Bimal Krishna Matilal*. Ed. by M. Chakraborty, B. Löwe, M. Nath Mitra, and S. Sarukkai. London: College Publications.

Bjerva, Johannes, Johan Bos, Rob Van der Goot, and Malvina Nissim (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)*, pp. 642–646.

Bos, Johan, Fabio Massimo Zanzotto, and Marco Pennacchiotti (2009). “Textual entailment at EVALITA 2009”. In: *Proceedings of EVALITA* 2009.6.4, p. 2.

Bowman, Samuel R. and George E. Dahl (2021). “What Will it Take to Fix Benchmarking in Natural Language Understanding?” In: *NAACL*.

Bowman, Samuel R., Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler (2020). “New Protocols and Negative Results for Textual Entailment Data Collection”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 8203–8214. DOI: 10.18653/v1/2020.emnlp-main.658. URL: https://www.aclweb.org/anthology/2020.emnlp-main.658.

Bowman, Samuel R, Gabor Angeli, Christopher Potts, and Christopher D Manning (2015). “A Large Annotated Corpus for Learning Natural Language Inference”. In: *Proceedings of EMNLP*. URL: https://www.aclweb.org/anthology/D15-1075/.

Budur, Emrah, Rıza Özçelik, Tunga Gungor, and Christopher Potts (2020). “Data and Representation for Turkish Natural Language Inference”. In: *Proceedings of EMNLP*. Online: Association for Computational Linguistics. URL: https://arxiv.org/pdf/2004.14963.pdf.

Carpenter, Bob (1998). *Type-Logical Semantics*. MIT Press.

Che, Wanxiang, Yunlong Feng, Libo Qin, and Ting Liu (2020). "N-LTP: A Open-source Neural Chinese Language Technology Platform with Pretrained Models". In: *arXiv preprint arXiv:2009.11616*.

Chen, Eric, Bert Gao, and Lawrence S. Moss (2021). "NeuralLog: Natural Language Inference with Joint Neural and Logical Reasoning". In: *Proceedings of *SEM*.

Chen, Jifan, Eunsol Choi, and Greg Durrett (2021). "Can NLI Models Verify QA Systems' Predictions?" In: *arXiv preprint arXiv:2104.08731*.

Chen, Qian, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen (2017). "Enhanced LSTM for Natural Language Inference". In: *Proceedings of ACL*. URL: https://arxiv.org/pdf/1609.06038.

Chen, Tongfei, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme (2020). "Uncertain Natural Language Inference". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8772–8779. DOI: 10.18653/v1/2020.acl-main.774. URL: https://www.aclweb.org/anthology/2020.acl-main.774.

Chen, Zeming and Qiyue Gao (2021). "Monotonicity Marking from Universal Dependency Trees". In: *Proceedings of IWCS*. URL: https://arxiv.org/abs/2104.08659.

Cheung, Jackie and Gerald Penn (2012). "Unsupervised detection of downward-entailing operators by maximizing classification certainty". In: *Proc. 13th EACL*, pp. 696–705.

Chierchia, Gennaro (2013). *Logic in grammar: Polarity, free choice, and intervention*. Oxford University Press.

Choi, Hyunjin, Judong Kim, Seongho Joe, Seungjai Min, and Youngjune Gwon (2021). *Analyzing Zero-shot Cross-lingual Transfer in Supervised NLP Tasks*. arXiv: 2101.10649 [cs.CL].

Church, Kenneth Ward and Patrick Hanks (1990). "Word association norms, mutual information, and lexicography". In: *Computational Linguistics* 16.1, pp. 22–29.

Clark, Kevin, Minh-Thang Luong, Quoc V Le, and Christopher D Manning (2019). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: *International Conference on Learning Representations*.

Clark, Peter, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz (2020). "From 'F' to 'A' on the N.Y. Regents Science Exams: An Overview of the Aristo Project". In: *AI Magazine* 41.4, pp. 39–53. DOI: 10.1609/aimag.v41i4.5304. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/5304.

Clark, Stephen and James R Curran (2007). "Wide-coverage efficient statistical parsing with CCG and log-linear models". In: *Computational Linguistics* 33.4, pp. 493–552.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2020). "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://www.aclweb.org/anthology/2020.acl-main.747.

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of EMNLP*. URL: https://www.aclweb.org/anthology/D17-1070.

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni (2018a). "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2126–2136. DOI: 10.18653/v1/P18-1198. URL: https://www.aclweb.org/anthology/P18-1198.

Conneau, Alexis and Guillaume Lample (2019). “Cross-lingual Language Model Pretraining”. In: *Advances in Neural Information Processing Systems*, pp. 7057–7067.

Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov (2018b). “XNLI: Evaluating Cross-lingual Sentence Representations”. In: *Proceedings of EMNLP*. URL: https://arxiv.org/abs/1809.05053.

Cooper, Robin, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. (1996). *Using the framework*. Tech. rep. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu (2020). “Revisiting Pre-Trained Models for Chinese Natural Language Processing”. In: *arXiv preprint arXiv:2004.13922*.

Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu (2019). “Pre-Training with Whole Word Masking for Chinese BERT”. In: *arXiv preprint arXiv:1906.08101*. URL: https://arxiv.org/abs/1906.08101.

Dagan, Ido, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava (2006). “Direct Word Sense Matching for Lexical Substitution”. In: *Proceedings of ACL*, pp. 449–456. DOI: 10.3115/1220175.1220232. URL: https://www.aclweb.org/anthology/P06-1057.

Dagan, Ido, Oren Glickman, and Bernardo Magnini (2005). “The PASCAL Recognizing Textual Entailment Challenge”. In: *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Danescu, Cristian and Lillian Lee (2010). “Don’t ‘have a clue’? Unsupervised co-learning of downward-entailing operators”. In: *Proceedings of ACL*.

Danescu, Cristian, Lillian Lee, and Richard Ducott (2009). “Without a ‘doubt’? Unsupervised discovery of downward-entailing operators”. In: *Proceedings of NAACL*.

De Marneffe, Marie-Catherine, Mandy Simons, and Judith Tonhauser (2019). "The CommitmentBank: Investigating projection in naturally occurring discourse". In: *Proceedings of Sinn und Bedeutung*. Vol. 23, pp. 107–124.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407.

Deng, Dun, Fenrong Liu, Mingming Liu, and Dag Westerståhl, eds. (2020). *Monotonicity in Logic and Language*. Springer.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of NAACL*. DOI: 10.18653/v1/N19-1423.

Dong, Yubing, Ran Tian, and Yusuke Miyao (2014). "Encoding generalized quantifiers in dependency-based compositional semantics". In: *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*.

Dowty, David (1994). "The Role of Negative Polarity and Concord Marking in Natural Language Reasoning". In: *Proceedings of Semantics and Linguistic Theory (SALT) IV*.

Dzikovska, Myroslava O, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang (2013). "SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge". In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 263–274.

Eichler, Kathrin, Aleksandra Gabryszak, and Günter Neumann (2014). "An analysis of textual inference in German customer emails". In: *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*, pp. 69–74.

Ethayarajh, Kawin (2019). "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings". In: *Proceed-*

*ings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 55–65. DOI: 10.18653/v1/D19-1006. URL: https://www.aclweb.org/anthology/D19-1006.

Ettinger, Allyson, Sudha Rao, Hal Daumé III, and Emily M Bender (2017). "Towards Linguistically Generalizable NLP Systems: A Workshop and Shared Task". In: *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pp. 1–10.

Falke, Tobias, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych (2019). "Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2214–2220. DOI: 10.18653/v1/P19-1213. URL: https://www.aclweb.org/anthology/P19-1213.

Feng, Shi, Eric Wallace, and Jordan Boyd-Graber (2019). "Misleading Failures of Partial-input Baselines". In: *Proceedings of ACL*. URL: https://arxiv.org/abs/1905.05778.

Firth, John R. (1957). *Papers in Linguistics*. London: Oxford University Press.

Fitting, Melvin (1990). *First-order logic and automated theorem proving*. Springer-Verlag.

Fonseca, E, L Santos, Marcelo Criscuolo, and S Aluisio (2016). "ASSIN: Avaliacao de similaridade semantica e inferencia textual". In: *Computational Processing of the Portuguese Language-12th International Conference*, pp. 13–15. URL: http://propor2016.di.fc.ul.pt/wp-content/uploads/2015/10/assin-overview.pdf.

Geiger, Atticus, Kyle Richardson, and Christopher Potts (2020). "Neural Natural Language Inference Models Partially Embed Theories of Lexical Entailment and Negation". In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 163–173.

DOI: 10.18653/v1/2020.blackboxnlp-1.16. URL: https://www.aclweb.org/anthology/2020.blackboxnlp-1.16.

Geva, Mor, Yoav Goldberg, and Jonathan Berant (2019). "Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets". In: *Proceedings of EMNLP-IJCNLP*. URL: https://arxiv.org/abs/1908.07898.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and William B Dolan (2007). "The third pascal recognizing textual entailment challenge". In: *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9.

Glockner, Max, Vered Shwartz, and Yoav Goldberg (2018). "Breaking NLI Systems with Sentences that Require Simple Lexical Inferences". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 650–655.

Goldberg, Yoav (2016). "A primer on neural network models for natural language processing". In: *Journal of Artificial Intelligence Research* 57, pp. 345–420.

Goodwin, Emily, Koustuv Sinha, and Timothy O'Donnell (2020). "Probing Linguistic Systematicity". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1958–1969.

Goyal, Naman, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau (2021). *Larger-Scale Transformers for Multilingual Masked Language Modeling*. arXiv: 2105.00572 [cs.CL].

Guo, Chuan, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten (2020). "Certified Data Removal from Machine Learning Models". In: *International Conference on Machine Learning*. PMLR, pp. 3832–3842. URL: https://arxiv.org/pdf/1911.03030.pdf.

Gururangan, Suchin, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith (2018). "Annotation Artifacts in Natural Language Inference Data". In: *Proceedings of NAACL*. URL: https://arxiv.org/pdf/1803.02324.

Haim, Roy Bar, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor (2006). "The II PASCAL RTE challenge". In: *PASCAL Challenges Workshop*.

Harabagiu, Sanda and Andrew Hickl (2006). "Methods for using textual entailment in open-domain question answering". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 905–912.

Hardalov, Momchil, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov (2020). "EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 5427–5444. DOI: 10.18653/v1/2020.emnlp-main.438. URL: https://www.aclweb.org/anthology/2020.emnlp-main.438.

Hayashibe, Yuta (2020). "Japanese Realistic Textual Entailment Corpus". In: *Proceedings of LREC*. URL: https://www.aclweb.org/anthology/2020.lrec-1.843/.

Hockenmaier, Julia and Mark Steedman (2007). "CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank". In: *Computational Linguistics* 33.3, pp. 355–396.

Honnibal, Matthew, James R. Curran, and Johan Bos (2010). "Rebanking CCGbank for Improved NP Interpretation". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 207–215. URL: https://www.aclweb.org/anthology/P10-1022.

Horn Laurence, R (1972). "On the semantic properties of logical operators in English". PhD thesis. University of California Los Angeles.

Hu, Hai, Qi Chen, and Lawrence S. Moss (2019). "Natural Language Inference with Monotonicity". In: *Proceedings of IWCS*.

Hu, Hai, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kübler (2020a). "MonaLog: a Lightweight System for Natural Language Inference Based on Monotonicity". In: *Proceedings of SCiL*. URL: https://arxiv.org/abs/1910.08772.

Hu, Hai and Sandra Kübler (2020). "Investigating Translated Chinese and Its Variants Using Machine Learning". In: *Natural Language Engineering (Special Issue on NLP for Similar Languages, Varieties and Dialects)*, pp. 1–34. DOI: 10.1017/S1351324920000182.

Hu, Hai, Wen Li, and Sandra Kübler (2018). "Detecting Syntactic Features of Translated Chinese". In: *Proceedings of the 2nd Workshop on Stylistic Variation*, pp. 20–28. URL: https://www.aclweb.org/anthology/W18-1603.

Hu, Hai and Lawrence S. Moss (2018). "Polarity Computations in Flexible Categorial Grammar". In: *Proceedings of *SEM*, pp. 124–129.

— (2020). "An Automatic Monotonicity Annotation Tool Based on CCG Trees". In: *Second Tsinghua Interdisciplinary Workshop on Logic, Language, and Meaning: Monotonicity in Logic and Language*.

Hu, Hai, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S. Moss (2020b). "OCNLI: Original Chinese Natural Language Inference". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 3512–3526. DOI: 10.18653/v1/2020.findings-emnlp.314. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.314.

Hu, Hai, He Zhou, Zuoyu Tian, Yiwen Zhang, Yina Ma, Yanting Li, Yixin Nie, and Kyle Richardson (2021). "Investigating Transfer Learning in Multilingual Pre-trained Language Models through Chinese Natural Language Inference". In: *Findings of ACL*.

Hu, Junjie, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson (2020). "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation". In: *Proceedings of the 37th International Confer-*

*ence on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4411–4421. URL: http://proceedings.mlr.press/v119/hu20b.html.

Huang, Yi Ting, Elizabeth Spelke, and Jesse Snedeker (2013). "What exactly do numbers mean?" In: *Language Learning and Development* 9.2, pp. 105–129.

Icard, Thomas F. (2012). "Inclusion and Exclusion in Natural Language". In: *Studia Logica* 100.4, pp. 705–725.

Icard, Thomas F. and Lawrence S. Moss (2013). "A Complete Calculus of Monotone and Antitone Higher-Order Functions". In: *Proceedings of TACL 2013*. Vol. 23. EPiC Series. Vanderbilt University, pp. 96–100.

— (2014). "Recent Progress on Monotonicity". In: *Linguistic Issues in Language Technology* 9.7, pp. 167–194.

Jia, Robin and Percy Liang (2017). "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031.

Joshi, Pratik, Somak Aditya, Aalok Sathe, and Monojit Choudhury (2020). "TaxiNLI: Taking a ride up the NLU hill". In: *arXiv preprint arXiv:2009.14505*.

Kalouli, Aikaterini-Lida, Richard Crouch, and Valeria de Paiva (2020). "Hy-NLI: a Hybrid system for Natural Language Inference". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 5235–5249. DOI: 10.18653/v1/2020.coling-main.459. URL: https://www.aclweb.org/anthology/2020.coling-main.459.

Kalouli, Aikaterini-Lida, Valeria de Paiva, and Livy Real (2017a). "Correcting contradictions". In: *Proceedings of the Computing Natural Language Inference Workshop*.

Kalouli, Aikaterini-Lida, Livy Real, and Valeria de Paiva (2017b). "Textual Inference: getting logic from humans". In: *Proceedings of IWCS*.

Kalouli, Aikaterini-Lida, Livy Real, and Valeria de Paiva (2018). "WordNet for "Easy" Textual Inferences". In: *Proceedings of LREC*. Miyazaki, Japan. ISBN: 979-10-95546-28-3.

Karthikeyan, K, Zihan Wang, Stephen Mayhew, and Dan Roth (2019). "Cross-Lingual Ability of Multilingual BERT: An Empirical Study". In: *International Conference on Learning Representations*.

Keenan, Edward L. and Leonard M. Faltz (1984). *Boolean Semantics for Natural Language*. Springer.

Kennedy, Christopher (2013). "A scalar semantics for scalar readings of number words". In: *From grammar to meaning: The spontaneous logicality of language* 172, p. 200.

Khashabi, Daniel, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, et al. (2020). "ParsiNLU: A Suite of Language Understanding Challenges for Persian". In: *arXiv preprint arXiv:2012.06154*.

Khot, Tushar, Ashish Sabharwal, and Peter Clark (2018). "SciTaiL: A Textual Entailment Dataset from Science Question Answering." In: *AAAI*. URL: https://ai2-website.s3.amazonaws.com/publications/scitail-aaai-2018_cameraready.pdf.

Kiela, Douwe, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams (2021). "Dynabench: Rethinking Benchmarking in NLP". In: *NAACL*.

Kim, Najoung, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick (2019). "Probing What Different NLP Tasks Teach Machines about Function Word Comprehension". In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for

Computational Linguistics, pp. 235–249. DOI: 10.18653/v1/S19-1026. URL: https://www.aclweb.org/anthology/S19-1026.

Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). "Skip-Thought Vectors". In: *NIPS*.

Koppel, Moshe and Noam Ordan (2011). "Translationese and its dialects". In: *Proceedings of ACL*. URL: https://dl.acm.org/doi/10.5555/2002472.2002636.

Kudo, Taku (2018). "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75.

Lample, Guillaume and Alexis Conneau (2019). "Cross-lingual language model pretraining". In: *NIPS*.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2019). "ALBERT: A lite BERT for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942*.

Lavalle-Martínez, José-de-Jesús, Manuel Montes-y-Gómez, Luis Villaseñor-Pineda, Héctor Jiménez-Salazar, and Ismael-Everardo Bárcenas-Patiño (2017). "Equivalences Among Polarity Algorithms". In: *Studia Logica*. ISSN: 1572-8730. DOI: 10.1007/s11225-017-9743-y. URL: https://doi.org/10.1007/s11225-017-9743-y.

Le Bras, Ronan, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E Peters, Ashish Sabharwal, and Yejin Choi (2020). "Adversarial Filters of Dataset Biases". In: *Proceedings of ICLR*. URL: https://arxiv.org/abs/2002.04108.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2020). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4, pp. 1234–1240.

Lewis, Mike, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer (2020). *Pre-training via Paraphrasing*. arXiv: 2006.15020 [cs.CL].

Lewis, Mike and Mark Steedman (2013). “Combined distributional and logical semantics”. In: *TACL* 1, pp. 179–192.

— (2014). “A* CCG parsing with a supertag-factored model”. In: *Proceedings of EMNLP*, pp. 990–1000.

Lewis, Patrick, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk (2020). “MLQA: Evaluating Cross-lingual Extractive Question Answering”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7315–7330.

Li, Charles N and Sandra A Thompson (1981). *Mandarin Chinese: A Functional Reference Grammar*. Univ of California Press.

Li, Haoran, Junnan Zhu, Jiajun Zhang, and Chengqing Zong (2018). “Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1430–1441. URL: https://www.aclweb.org/anthology/C18-1121.

Li, Shen, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du (2018). “Analogical Reasoning on Chinese Morphological and Semantic Relations”. In: *Proceedings of ACL*. URL: http://aclweb.org/anthology/P18-2023.

Lin, Chien-Jer Charles (2011). “Chinese and English relative clauses: Processing constraints and typological consequences”. In: *Proceedings of the 23rd North American Conference on Chinese Linguistics (NACCL-23)*, pp. 191–199. URL: http://www.naccl.osu.edu/sites/naccl.osu.edu/files/NACCL-23_1_13.pdf.

Lin, Yi-Chung and Keh-Yih Su (2021). “How Fast can BERT Learn Simple Natural Language Inference?” In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 626–633. URL: https://www.aclweb.org/anthology/2021.eacl-main.51.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer (2020). "Multilingual denoising pre-training for neural machine translation". In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692*. URL: https://arxiv.org/abs/1907.11692.

Longpre, Shayne, Yi Lu, and Joachim Daiber (2020). *MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering*. arXiv: 2007.15207 [cs.CL].

MacCartney, Bill (2009). "Natural Language Inference". PhD thesis. Stanford University.

MacCartney, Bill and Christopher D Manning (2008). "Modeling semantic containment and exclusion in natural language inference". In: *Proceedings of COLING*. Association for Computational Linguistics, pp. 521–528. URL: http://www.aclweb.org/anthology/C08-1066s.

— (2007). "Natural Logic for Textual Inference". In: *ACL Workshop on Textual Entailment and Paraphrasing*, pp. 193–200.

— (2009). "An extended model of natural logic". In: *Proceedings of IWCS*, pp. 140–156.

Manning, Christopher D (2006). *Local textual inference: it's hard to circumscribe, but you know it when you see it–and NLP needs it*. unpublished manuscript.

— (2011). "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" In: *International conference on intelligent text processing and computational linguistics*. Springer, pp. 171–189.

Marelli, M., S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli (2014). "A SICK cure for the evaluation of compositional distributional semantic models". In:

*Proceedings of LREC*. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

Marelli, Marco, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli (2014). "SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1–8. DOI: 10.3115/v1/S14-2001. URL: https://www.aclweb.org/anthology/S14-2001.

de Marneffe, Marie-Catherine, Anna N. Rafferty, and Christopher D. Manning (2008). "Finding Contradictions in Text". In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 1039–1047. URL: https://www.aclweb.org/anthology/P08-1118.

Martínez-Gómez, Pascual, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki (2016). "ccg2lambda: A Compositional Semantics System". In: *Proceedings of ACL: System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, pp. 85–90. URL: https://aclweb.org/anthology/P/P16/P16-4015.pdf.

— (2017). "On-demand injection of lexical knowledge for recognising textual entailment". In: *Proceedings of EACL*. URL: https://www.aclweb.org/anthology/E17-1067.

Marvin, Rebecca and Tal Linzen (2018). "Targeted Syntactic Evaluation of Language Models". In: *Proceedings of EMNLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 1192–1202. DOI: 10.18653/v1/D18-1151. URL: https://www.aclweb.org/anthology/D18-1151.

McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher (2017). "Learned in translation: contextualized word vectors". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6297–6308.

McCoy, R Thomas, Ellie Pavlick, and Tal Linzen (2019). “Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference”. In: *Proceedings of ACL*. URL: https://www.aclweb.org/anthology/P19-1334.

McEnery, Anthony and Zhonghua Xiao (2004). “The Lancaster Corpus of Mandarin Chinese: A Corpus for Monolingual and Contrastive Language Study”. In: *LREC*, pp. 1175–1178. URL: http://www.lrec-conf.org/proceedings/lrec2004/summaries/231.htm.

Mehdad, Yashar, Matteo Negri, and Marcello Federico (2011). “Using Bilingual Parallel Corpora for Cross-lingual Textual Entailment”. In: *Proceedings of ACL*. URL: https://www.aclweb.org/anthology/P11-1134/.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Efficient Estimation of Word Representations in Vector Space”. In: *Proceedings of International Conference on Learning Representations (ICLR)*. Scottsdale, AZ.

Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

Mineshima, Koji, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki (2015). “Higher-order logical inference with compositional semantics”. In: *Proceedings of Empirical Methods in Natural Language Processing*, pp. 2055–2061.

Moss, Lawrence S. (2012). “The Soundness of Internalized Polarity Marking”. In: *Studia Logica* 100.4, pp. 683–704.

— (2018). *Lecture notes on Logic*.

Mou, Lili, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin (2016). “Natural Language Inference by Tree-Based Convolution and Heuristic Matching”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 130–136. DOI: 10.18653/v1/P16-2022. URL: https://www.aclweb.org/anthology/P16-2022.

Musolino, Julien (2004). “The semantics and acquisition of number words: Integrating linguistic and developmental perspectives”. In: *Cognition* 93.1, pp. 1–41.

Musolino, Julien (2009). “The logical syntax of number words: Theory, acquisition and processing”. In: *Cognition* 111.1, pp. 24–45.

Naik, Aakanksha, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig (2018). “Stress Test Evaluation for Natural Language Inference”. In: *Proceedings of COLING*. URL: https://www.aclweb.org/anthology/C18-1198.pdf.

Nairn, Rowan, Cleo Condoravdi, and Lauri Karttunen (2006). “Computing relative polarity for textual inference”. In: *Proceedings of ICoS-5 (Inference in Computational Semantics)*. Buxton, UK. URL: http://www.aclweb.org/anthology/W06-3907.

Nangia, Nikita and Samuel Bowman (2019). “Human vs. Muppet: A Conservative Estimate of Human Performance on the GLUE Benchmark”. In: *Proceedings of ACL*. URL: https://arxiv.org/abs/1905.10425.

Negri, Matteo, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti (2011). “Divide and Conquer: Crowdsourcing the Creation of Cross-lingual Textual Entailment Corpora”. In: *Proceedings of EMNLP*. URL: https://www.aclweb.org/anthology/D11-1062/.

Nie, Yixin and Mohit Bansal (2017). “Shortcut-Stacked Sentence Encoders for Multi-Domain Inference”. In: *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pp. 41–45.

Nie, Yixin, Haonan Chen, and Mohit Bansal (2019). “Combining Fact Extraction and Verification with Neural Semantic Matching Networks”. In: *Association for the Advancement of Artificial Intelligence (AAAI)*.

Nie, Yixin, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela (2020a). “Adversarial NLI: A New Benchmark for Natural Language Understanding”. In: *Proceedings of ACL*. URL: https://www.aclweb.org/anthology/2020.acl-main.441.

Nie, Yixin, Xiang Zhou, and Mohit Bansal (2020b). “What Can We Learn from Collective Human Opinions on Natural Language Inference Data?” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online:

Association for Computational Linguistics, pp. 9131–9143. DOI: 10.18653/v1/2020.emnlp-main.734. URL: https://www.aclweb.org/anthology/2020.emnlp-main.734.

Nozza, Debora, Federico Bianchi, and Dirk Hovy (2020). "What the [mask]? Making sense of language-specific BERT models". In: *arXiv preprint arXiv:2003.02912*.

Parrish, Alicia, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alex Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R Bowman (2021). "Does Putting a Linguist in the Loop Improve NLU Data Collection?" In: *arXiv preprint arXiv:2104.07179*.

Pasunuru, Ramakanth, Han Guo, and Mohit Bansal (2017). "Towards Improving Abstractive Summarization via Entailment Generation". In: *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 27–32. DOI: 10.18653/v1/W17-4504. URL: https://www.aclweb.org/anthology/W17-4504.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems* 32, pp. 8026–8037.

Pavlick, Ellie and Chris Callison-Burch (2016). "Most "babies" are "little" and most "problems" are "huge": Compositional Entailment in Adjective-Nouns". In: *Proceedings of ACL*, pp. 2164–2173.

Pavlick, Ellie and Tom Kwiatkowski (2019). "Inherent Disagreements in Human Textual Inferences". In: *Transactions of the Association for Computational Linguistics* 7, pp. 677–694. URL: https://transacl.org/index.php/tacl/article/view/1780.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of EMNLP*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep Contextualized Word Representations". In: *Proceedings of NAACL*. URL: https://arxiv.org/abs/1802.05365.

Phang, Jason, Iacer Calixto, Phu Mon Htut, Yada Pruksachatkun, Haokun Liu, Clara Vania, Katharina Kann, and Samuel R. Bowman (2020). "English Intermediate-Task Training Improves Zero-Shot Cross-Lingual Transfer Too". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 557–575. URL: https://www.aclweb.org/anthology/2020.aacl-main.56.

Phang, Jason, Thibault Févry, and Samuel R Bowman (2018). "Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks". In: *arXiv preprint arXiv:1811.01088*.

Pires, Telmo, Eva Schlinger, and Dan Garrette (2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001.

Poliak, Adam (2020). "A survey on Recognizing Textual Entailment as an NLP Evaluation". In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pp. 92–109.

Poliak, Adam, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme (2018). "Hypothesis Only Baselines in Natural Language Inference". In: *Proceedings of *SEM*. URL: https://arxiv.org/pdf/1805.01042.

Ponti, Edoardo Maria, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen (2020). "XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2362–2376.

Potts, Christopher (2021). *Lecture notes for CS224u*. URL: http://web.stanford.edu/class/cs224u/slides/cs224u-2021-nli-part1-handout.pdf.

Potts, Christopher, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela (2020). "DynaSent: A Dynamic Benchmark for Sentiment Analysis". In: *arXiv preprint arXiv:2012.15349*. URL: https://arxiv.org/abs/2012.15349.

Pruksachatkun, Yada, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman (2020). "Intermediate Task Transfer Learning with Pretrained Language Models: When and Why Does It Work?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5231–5247. DOI: 10.18653/v1/2020.acl-main.467. URL: https://www.aclweb.org/anthology/2020.acl-main.467.

Real, Livy, Erick Fonseca, and Hugo Gonçalo Oliveira (2020). "Organizing the ASSIN 2 Shared Task". In: *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese*.

Real, Livy, Ana Rodrigues, Andressa Vieira e Silva, Beatriz Albiero, Bruna Thalenberg, Bruno Guide, Cindy Silva, Guilherme de Oliveira Lima, Igor C. S. Câmara, Miloš Stanojević, Rodrigo Souza, and Valeria de Paiva (2018). "SICK-BR: A Portuguese Corpus for Inference". In: *Computational Processing of the Portuguese Language*. Ed. by Aline Villavicencio, Viviane Moreira, Alberto Abad, Helena Caseli, Pablo Gamallo, Carlos Ramisch, Hugo Gonçalo Oliveira, and Gustavo Henrique Paetzold. Springer, pp. 303–312. ISBN: 978-3-319-99722-3.

Reif, Emily, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim (2019). "Visualizing and measuring the geometry of BERT". In: *Advances in Neural Information Processing Systems* 32, pp. 8594–8603.

Reimers, Nils and Iryna Gurevych (2019). “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of EMNLP*. URL: https://arxiv.org/abs/1908.10084.

Ribeiro, Marco Tulio, Tongshuang Wu, Carlos Guestrin, and Sameer Singh (2020). “Beyond Accuracy: Behavioral Testing of NLP models with CheckList”. In: *Association for Computational Linguistics (ACL)*.

Richardson, Kyle, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal (2020). “Probing Natural Language Inference Models through Semantic Fragments”. In: *Proceedings of AAAI*. URL: https://arxiv.org/abs/1909.07521.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2020). “A Primer in BERTology: What We Know About How BERT Works”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866. DOI: 10.1162/tacl_a_00349. URL: https://www.aclweb.org/anthology/2020.tacl-1.54.

S. Moss, Lawrence and Hai Hu (in prep). *Automatic Polarity Tagging Using CCG Trees*.

Saha, Swarnadeep, Yixin Nie, and Mohit Bansal (2020). “ConjNLI: Natural Language Inference over Conjunctive Sentences”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8240–8252.

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi (2020). “WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale”. In: *Proceedings of AAAI*. URL: https://arxiv.org/abs/1907.10641.

Salvatore, Felipe, Marcelo Finger, and Roberto Hirata Jr (2019). “A logical-based corpus for cross-lingual evaluation”. In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pp. 22–30.

Sánchez-Valencia, Victor (1991). “Studies on Natural Logic and Categorial Grammar”. PhD thesis. Universiteit van Amsterdam.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://www.aclweb.org/anthology/P16-1162.

Siddhant, Aditya, Melvin Johnson, Henry Tsai, Naveen Arivazhagan, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman (2020). "Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation". In: *Proceedings of AAAI*. URL: https://arxiv.org/abs/1909.00437.

Sinha, Koustuv, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams (2020). "Unnatural Language Inference". In: *arXiv preprint arXiv:2101.00010*.

Song, Yan, Shuming Shi, Jing Li, and Haisong Zhang (2018). "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 175–180.

Steedman, Mark (2000). *The Syntactic Process*. Cambridge, MA: The MIT Press. ISBN: 0-262-19420-1.

Steinert-Threlkeld, Shane and Jakub Szymanik (2019). "Learnability and semantic universals". In: *Semantics and Pragmatics* 12, p. 4.

Subramanian, Sandeep, Adam Trischler, Yoshua Bengio, and Christopher J Pal (2018). "Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning". In: *Proceedings of ICLR*. URL: https://arxiv.org/abs/1804.00079.

Sun, Tianqi (2009). “A Study on the License Pattern and Mechanism of Non-core Arguments in Mandarin Chinese”. PhD thesis. Peking University.

Sung, Chul, Tejas Indulal Dhamecha, and Nirmal Mukhi (2019). “Improving short answer grading using transformer-based pre-training”. In: *International Conference on Artificial Intelligence in Education*. Springer, pp. 469–481.

Taylor, Wilson L (1953). ““Cloze procedure”: A new tool for measuring readability”. In: *Journalism quarterly* 30.4, pp. 415–433.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (2019a). “BERT Rediscovers the Classical NLP Pipeline”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601.

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, and Ellie Pavlick (2019b). “What do you learn from context? Probing for sentence structure in contextualized word representations”. In: *International Conference on Learning Representations*.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal (2018). “FEVER: a Large-scale Dataset for Fact Extraction and VERification”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819.

Tian, Ran, Yusuke Miyao, and Takuya Matsuzaki (2014). “Logical inference on dependency-based compositional semantics”. In: *Proceedings of ACL*. Vol. 1, pp. 79–89.

Trivedi, Harsh, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian (2019). “Repurposing Entailment for Multi-Hop Question Answering Tasks”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguis-

tics, pp. 2948–2958. DOI: 10.18653/v1/N19-1302. URL: https://www.aclweb.org/anthology/N19-1302.

Tsuchiya, Masatoshi (2018). "Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Turney, Peter D and Patrick Pantel (2010). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37, pp. 141–188.

Uppal, Shagun, Vivek Gupta, Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn Shah, and Amanda Stent (2020). "Two-Step Classification using Recasted Data for Low Resource Settings". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 706–719. URL: https://www.aclweb.org/anthology/2020.aacl-main.71.

Vania, Clara, Ruijie Chen, and Samuel R. Bowman (2020). "Asking Crowdworkers to Write Entailment Examples: The Best of Bad Options". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 672–686. URL: https://www.aclweb.org/anthology/2020.aacl-main.68.

Vashishtha, Siddharth, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White (2020). "Temporal Reasoning in Natural Language Inference". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4070–4078. DOI: 10.18653/v1/2020.findings-emnlp.363. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.363.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *NIPS*.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). "SuperGLUE: A Stickier Benchmark for General-purpose Language Understanding Systems". In: *Advances in Neural Information Processing Systems*. URL: http://papers.nips.cc/paper/8589-superglue-a-stickier-benchmark-for-general-purpose-language-understanding-systems.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of BlackboxNLP*. URL: https://arxiv.org/abs/1804.07461.

Wang, Sinong, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma (2021). *Entailment as Few-Shot Learner*. arXiv: 2104.14690 [cs.CL].

Wheatley, Barbara (1996). *CALLHOME Mandarin Chinese Transcripts LDC96T16*.

Wijnholds, Gijs and Michael Moortgat (2021). "SICKNL: A Dataset for Dutch Natural Language Inference". In: *Proceedings of EACL*. URL: https://arxiv.org/abs/2101.05716.

Williams, Adina, Nikita Nangia, and Samuel R Bowman (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Williams, Adina, Tristan Thrush, and Douwe Kiela (2020). "ANLIzing the Adversarial Natural Language Inference Dataset". In: *arXiv preprint arXiv:2010.12729*. URL: https://arxiv.org/abs/2010.12729.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam

Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush (2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Shijie and Mark Dredze (2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://www.aclweb.org/anthology/D19-1077.

— (2020). "Are All Languages Created Equal in Multilingual BERT?" In: *arXiv preprint arXiv:2005.09093*.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144*.

Xu, Liang, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan (2020). "CLUE: A Chinese Language Understanding Evaluation Benchmark". In: *Proceedings of COLING*. URL: https://arxiv.org/abs/2004.05986.

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (2020). "mT5: A massively multilingual pre-trained text-to-text transformer". In: *arXiv preprint arXiv:2010.11934*.

Xun, Endong, Gaoqi Rao, Xiaoyue Xiao, and Jiaojiao Zang (2016). "The construction of the BCC Corpus in the age of Big Data". In: *Corpur Linguistics (in Chinese)*, pp. 93–109.

Yanaka, Hitomi, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos (2019a). "Can neural networks understand monotonicity reasoning?" In: *Proceedings of BlackboxNLP*.

— (2019b). "HELP: A Dataset for Identifying Shortcomings of Neural Models in Monotonicity Reasoning". In: *Proceedings of *SEM*, pp. 250–255.

Yanaka, Hitomi, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki (2018). "Acquisition of Phrase Correspondences Using Natural Deduction Proofs". In: *Proceedings of NAACL*. URL: https://www.aclweb.org/anthology/N18-1069/.

Yang, Yinfei, Yuan Zhang, Chris Tar, and Jason Baldridge (2019). "PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification". In: *Proceedings of EMNLP*. Hong Kong, China: Association for Computational Linguistics, pp. 3685–3690. DOI: 10.18653/v1/D19-1382. URL: https://www.aclweb.org/anthology/D19-1382.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems* 32, pp. 5753–5763.

Yin, Wenpeng and Hinrich Schütze (2017). "Task-Specific Attentive Pooling of Phrase Alignments Contributes to Sentence Matching". In: *Proceedings of EACL*, pp. 699–709.

Yoshikawa, Masashi, Hiroshi Noji, and Yuji Matsumoto (2017). "A* CCG Parsing with a Supertag and Dependency Factored Model". In: *Proceedings of ACL*, pp. 277–287.

Zellers, Rowan, Yonatan Bisk, Roy Schwartz, and Yejin Choi (2018). “SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zeman, Daniel, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov (2018). “CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies”. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pp. 1–21.

Zhang, Sheng, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme (2017). “Ordinal common-sense inference”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 379–395.

Zhao, Wei, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu (2020). “Ape210K: A Large-Scale and Template-Rich Dataset of Math Word Problems”. In: *arXiv preprint arXiv:2009.11506*. URL: https://arxiv.org/abs/2009.11506.

Zhu, Yukun, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 19–27.

# CURRICULUM VITAE

Name: **Hai Hu**

Email address: *hu.hai@outlook.com*

Personal website: https://huhailinguist.github.io/

## EDUCATION

Ph.D., in Linguistics with a concentration in Computational Linguistics, minor in Cognitive Science — 2021
Indiana University

M.A. in English Linguistics — 2015
Renmin University of China

B.A. in English Language and Literature, minor in Chinese — 2012
Renmin University of China

## RESEARCH INTERESTS

Broad: Computational linguistics for syntax and semantics; cognitive science

Narrow: *Syntax*: using machine learning to study variation in Chinese (e.g., translational Chinese), treebank annotation, text classification based on stylistic features, filler-gap dependency. *Semantics*: modeling natural language inference, dataset creation and crowdsourcing, data augmentation using natural logic, word embeddings for semantic change.

## SELECTED GRANTS AND AWARDS

College of Arts and Sciences Dissertation Completion Fellowship, IU — 2020-2021

GPSG Research Award, IU — 2021

| | |
|---|---|
| Grant-in-Aid for Doctoral Research, IU | 2020 |
| Indiana Univ–Renmin Univ seed grant | 2019-2021 |
| GPSG Travel Award, IU | 2019 |
| IDAH Summer Incubator Award, IU | 2019 |
| Householder Research Award, IU Linguistics | 2019 |

## PUBLICATIONS


Hu, Hai, He Zhou, Zuoyu Tian, Yiwen Zhang, Yina Ma, Yanting Li, Yixin Nie, and Kyle Richardson (2021). "Investigating Transfer Learning in Multilingual Pre-trained Language Models through Chinese Natural Language Inference". In: *Findings of ACL*.

Hu, Hai, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kübler (2020a). "MonaLog: a Lightweight System for Natural Language Inference Based on Monotonicity". In: *Proceedings of SCiL*. URL: https://arxiv.org/abs/1910.08772.

Hu, Hai and Sandra Kübler (2020). "Investigating Translated Chinese and Its Variants Using Machine Learning". In: *Natural Language Engineering (Special Issue on NLP for Similar Languages, Varieties and Dialects)*, pp. 1–34. DOI: 10.1017/S1351324920000182.

Hu, Hai, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Sandra Kübler, and Chien-Jer Charles Lin (2020b). "Building a Treebank for Chinese Literature for Translation Studies". In: *Proceedings of 19th International Workshop on Treebanks and Linguistic Theories (TLT)*, pp. 18–31. URL: https://www.aclweb.org/anthology/2020.tlt-1.2.pdf.

Hu, Hai, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S. Moss (2020c). "OCNLI: Original Chinese Natural Language Inference". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computa-

tional Linguistics, pp. 3512–3526. DOI: 10.18653/v1/2020.findings-emnlp.314. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.314.

Li, Junyi, Hai Hu, Xuanwei Zhang, Minglei Li, Lu Li, and Liang Xu (2020). "Light Pre-Trained Chinese Language Model for NLP Tasks". In: *Natural Language Processing and Chinese Computing*. Ed. by Xiaodan Zhu, Min Zhang, Yu Hong, and Ruifang He. Springer International Publishing, pp. 567–578. ISBN: 978-3-030-60457-8.

Richardson, Kyle, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal (2020). "Probing Natural Language Inference Models through Semantic Fragments". In: *Proceedings of AAAI*. URL: https://arxiv.org/abs/1909.07521.

Xu, Liang, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan (2020). "CLUE: A Chinese Language Understanding Evaluation Benchmark". In: *Proceedings of COLING*. URL: https://arxiv.org/abs/2004.05986.

Hu, Hai, Qi Chen, and Lawrence S. Moss (2019). "Natural Language Inference with Monotonicity". In: *Proceedings of IWCS*.

Hu*, Hai, Wen Li*, He Zhou*, Zuoyu Tian, Yiwen Zhang, and Liang Zou (2019). "Ensemble Methods to Distinguish Mainland and Taiwan Chinese". In: *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pp. 165–171. URL: http://web.science.mq.edu.au/ smalmasi/vardial6/pdf/W19-1417.pdf. *: equal contributions.

Hu, Hai, Thomas F Icard, and Lawrence S. Moss (2018a). "Automated Reasoning from Polarized Parse Trees". In: *Proceedings of the Fifth Workshop on Natural Language and Computer Science*.

Hu, Hai, Wen Li, and Sandra Kübler (2018b). “Detecting Syntactic Features of Translated Chinese”. In: *Proceedings of the 2nd Workshop on Stylistic Variation*, pp. 20–28. URL: https://www.aclweb.org/anthology/W18-1603.

Hu, Hai and Lawrence S. Moss (2018). “Polarity Computations in Flexible Categorial Grammar”. In: *Proceedings of *SEM*, pp. 124–129.

Hu, Hai, Daniel Dakota, and Sandra Kübler (2017). “Non-Deterministic Segmentation for Chinese Lattice Parsing”. In: *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pp. 316–324.

Hu, Hai and Yiwen Zhang (2017). “Path of Vowel Raising in Chengdu Dialect of Mandarin”. In: *Proceedings of 29th North America Conference on Chinese Linguistics*, pp. 481–498.

Cavar, Damir, Lwin Moe, Hai Hu, and Kenneth Steimel (2016). “Preliminary Results from the Free Linguistic Environment Project”. In: *Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar (HeadLex 2016)*, pp. 161–181.

Hu, Hai, Patrícia Amaral, and Sandra Kübler (accepted). “Word Embeddings and Semantic Shifts in Historical Spanish: Methodological Considerations”. In: *Digital Scholarship in the Humanities*.

## TEACHING

| | |
|---|---|
| TA for Q520: Math and Logic for Cognitive Science | 2020 SP |
| Instructor for L203: Introduction to Linguistic Analysis | 2019 FA |
| TA for L103: Introduction to the Study of Language | 2017 FA |

## TRANSLATION

《h a与 , ( ——{ Ô 之 • Œ ô之 k》, ¯世¾/Q · W e/á w/H z Ñ Y _人 ú H > 2018

Chinese translation of *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*

by Douglas Hofstadter and Emmanuel Sander.

## SERVICE

*Reviewer:*

2021: ACL, NALOMA, CCL, EMNLP 2020: ACL, COLING, NALOMA

2019: SLSP, Journal of Natural Language Engineering, SIGMORPHON, NAACL

2018: COLING, EMNLP 2017: EMNLP, Indiana University Linguistic Club (IULC) Working Paper

*Organizer:*

Shared-task on Chinese NER and Anaphora Resolution using Light-weight models (https://www.cluebenchmarks.com/NLPCC.html), Session at NLPCC 2020.

NAtural LOgic meets MAchine Learning (NALOMA, https://typo.uni-konstanz.de/naloma20/), Workshop at WeSSLLI 2020.

## LANGUAGES

Native: Mandarin Chinese

Fluent: English

Beginner: German, Japanese

## PROGRAMMING SKILLS

Python, R, Jave, Unix, LaTeX